

**NANYANG
TECHNOLOGICAL
UNIVERSITY**
SINGAPORE

**AY2025/2026 Semester 1
SC4020 DATA ANALYTICS & MINING
Group 32**

Project 2

Matriculation Number	Student Name
U2340547A	Chow Jia Hui
U2340736F	Lim Pei Jun, Rena
U2320946A	Wonyeong Yoon

Table Of Contents

1.0 Task 1: Analysis of Symptom Co-occurrence Patterns	3
1.1 Introduction & Objective	3
1.2 Data Preprocessing	3
1.3 Apriori Algorithm Implementation	3
1.4 Parameter Selection	3
1.5 Results	4
2.0 Task 2: Mining Cancer Feature Patterns	5
2.1 Introduction & Objective	5
2.2 Data Preprocessing	5
2.3 Extracting Top-5 ‘high’ features	5
2.4 Results	6
2.5 Sensitivity Check	6
3.0 Task 3: EHR Data Synthesis with Machine Learning	7
3.1 Abstract	7
3.2 Objective	8
3.3 Background	8
3.3.1 Tabular Variational Autoencoder (TVAE)	8
3.3.2 Conditional Tabular Generative Adversarial Networks (CTGAN)	8
3.3.3 Gaussian Copula Variational Autoencoder (GCVAE)	9
3.4 Data Preparation	9
3.4.1 Dataset	9
3.4.2 Encoding Categorical Features	10
3.5 Results	10
References	12

1.0 Task 1: Analysis of Symptom Co-occurrence Patterns

1.1 Introduction & Objective

The objective of this task is to analyse co-occurrence patterns of symptoms within disease profiles using the Disease Symptom Prediction Dataset. By applying the Apriori algorithm, we aim to identify frequent combinations of symptoms that commonly appear together in the same diseases, which can provide valuable insights for clinical diagnosis and disease understanding.

1.2 Data Preprocessing

The dataset contains disease-symptom relationships with symptoms spread across 17 columns.

The preprocessing steps included:

1. **Data cleaning:** All symptom names are converted to lowercase and stripped of whitespace to ensure consistency. NaN values were also removed from the datasets.
2. **Data Transformation:** The dataset was melted to create disease-symptom pairs, where each disease represents a “basket” and its symptoms are the “items”.
3. **Transaction Encoding:** Symptoms were one-hot encoded to create a binary matrix (1 = symptom present, 0 = symptom absent) suitable for the Apriori algorithm

The final processed dataset contained 131 unique symptoms across multiple disease profiles.

1.3 Apriori Algorithm Implementation

The Apriori algorithm was implemented to extract frequent itemsets from our disease-symptom dataset and generate meaningful association rules. It operates on a foundational principle crucial for efficiency: if a specific set of symptoms is frequent across diseases, then every subset of that combination must also be frequent. This logic allows the algorithm to systematically build from a single symptom to larger combinations while pruning unlikely candidates early, making it highly effective for analysing our dataset [1].

The key components of the Apriori algorithm are:

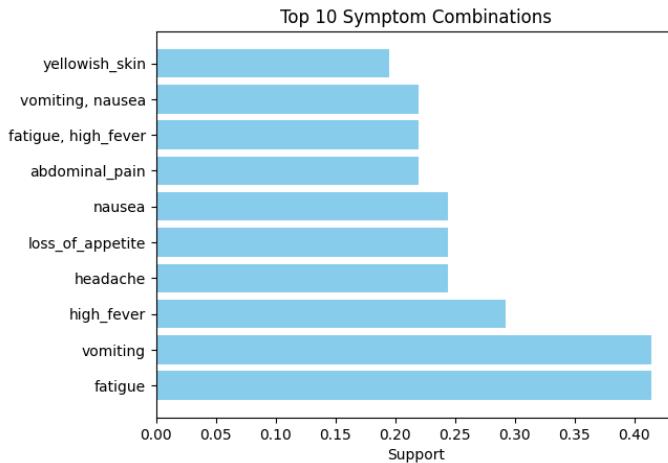
1. **Support:** The proportion of diseases that contain a specific symptom combination.
2. **Confidence:** The conditional probability that a disease having symptom A also has symptom B.
3. **Lift:** Measures how much more often symptoms A and B occur together than expected if they were statistically independent.

1.4 Parameter Selection

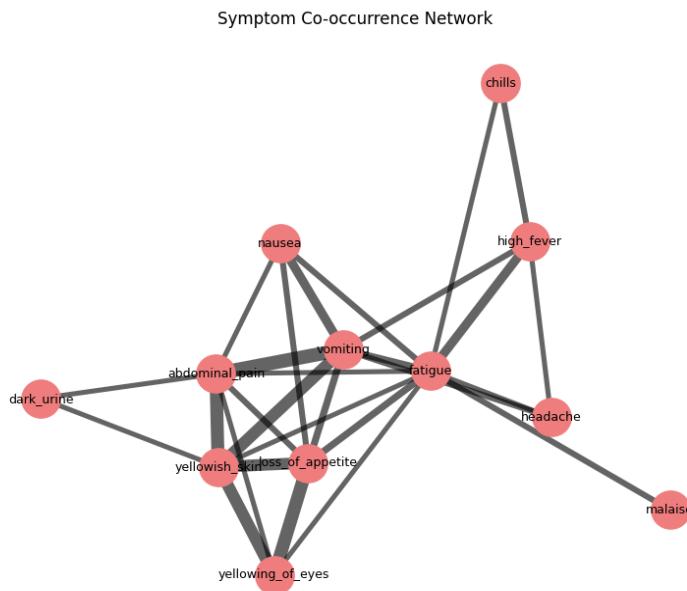
The following parameters are selected for the Apriori algorithm:

1. Minimum Support = 0.05, which means the symptoms must appear in at least 5% of the disease record to be considered frequent.
2. Metric used: Lift
3. Minimum threshold = 1.0, meaning we are finding positive association between two symptoms, where the presence of one symptom increases the likelihood of another symptom.

1.5 Results



The results show that single symptoms such as “fatigue” and “vomiting” have the highest individual frequencies. Among combinations, symptom pairs such as “fatigue, high_fever” and “vomiting, nausea” show strong co-occurrence patterns.



The network visualisation clearly illustrates symptoms that frequently co-occur together. A strong interconnected pathway is evident in the center of the graph, which includes symptoms

such as abdominal pain, vomiting, yellowish skin, loss of appetite, and yellowing of eyes. This dense clustering suggests a common underlying pathophysiology, potentially representing liver dysfunction or gastrointestinal disorder, where these symptoms are commonly present together in clinical cases.

2.0 Task 2: Mining Cancer Feature Patterns

2.1 Introduction & Objective

This task focuses on analysing feature sequences in the Breast Cancer Wisconsin Diagnostic dataset to identify patterns that distinguish malignant from benign cases. Using sequential pattern mining techniques, we transform continuous cancer features into categorical sequences to discover distinctive feature patterns associated with each diagnosis category.

2.2 Data Preprocessing

The dataset contains 30 continuous numerical features computed from digitised images of breast mass tissues. These continuous values are then transformed into categorical representations using KBinsDiscretizer function from scikit-learn to facilitate sequential pattern mining. A quantile binning strategy (equal-frequency bins) was employed with three bins and ordinal encoding {0, 1, 2}, which were subsequently mapped to categorical labels {0: low, 1: medium, 2: high}.

Following discretisation, feature sequences were generated for each patient using a ranking-based approach.

1. Feature ranking: After discretisation, features were ranked based on their categorical values.
2. Top-K selection: Only features categorised as ‘high’ were selected for sequence inclusion.
3. Sequence construction: The top 5 ‘high’-ranked features were ordered to form each patient’s sequence.
4. Diagnosis grouping: Sequences were separated into malignant (M) and benign (B) groups for pattern analysis.

2.3 Extracting Top-5 ‘high’ features

	diagnosis	sequence
0	M	[radius_mean, perimeter_mean, area_mean, smoot...
1	M	[radius_mean, perimeter_mean, area_mean, conca...

2	M	[radius_mean, texture_mean, perimeter_mean, ar...
3	M	[smoothness_mean, compactness_mean, concavity_...
4	M	[radius_mean, perimeter_mean, area_mean, compa...

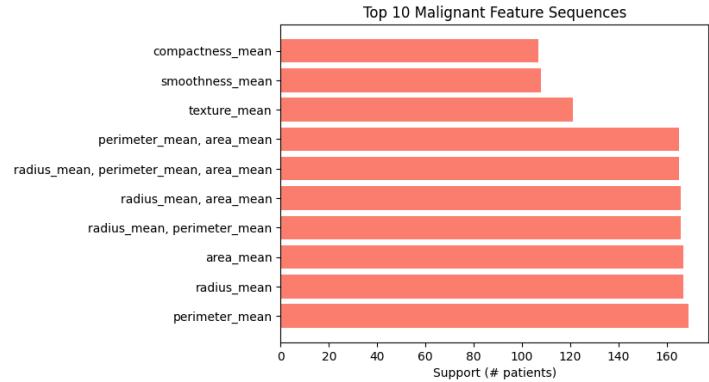
For each patient, we select the features whose binned value is classified as ‘high’. We keep at most the first five of these features and store them as that patient’s sequence.

2.4 Results

To better visualise the results, we first separate the sequences by diagnosis (Malignant vs Benign). Then using PrefixSpan, we identify the frequent subsequences of ‘high’ features that occur in multiple patients. Finally, we visualise the results by plotting the top 10 frequent sequences of ‘high’ features for the malignant patients.

```
Malignant Patterns:
['perimeter_mean'] (support=169)
['radius_mean'] (support=167)
['area_mean'] (support=167)
['radius_mean', 'perimeter_mean'] (support=166)
['radius_mean', 'area_mean'] (support=166)
['radius_mean', 'perimeter_mean', 'area_mean'] (support=165)
['perimeter_mean', 'area_mean'] (support=165)
['texture_mean'] (support=121)
['smoothness_mean'] (support=108)
['compactness_mean'] (support=107)

Benign Patterns:
['texture_se'] (support=113)
['fractal_dimension_mean'] (support=110)
['smoothness_se'] (support=97)
['symmetry_mean'] (support=83)
['smoothness_mean'] (support=82)
['symmetry_se'] (support=75)
['texture_mean'] (support=70)
['smoothness_mean', 'fractal_dimension_mean'] (support=54)
['fractal_dimension_mean', 'smoothness_se'] (support=47)
['texture_se', 'smoothness_se'] (support=47)
```



From the bar graph, we can see that size-related features including perimeter_mean, radius_mean and area_mean are the most frequently labelled as ‘high’ in malignant tumors. Although features such as compact_mean, smoothness_mean and texture_mean have lower support, they still occur frequently enough, indicating surface irregularity and roughness are commonly observed.

2.5 Sensitivity Check

A sensitivity check is performed to assess how our results will change when a different discretisation method is applied. Each binning strategy (quantile, uniform and kmeans) changes how the numeric features will be mapped to each low/medium/high.

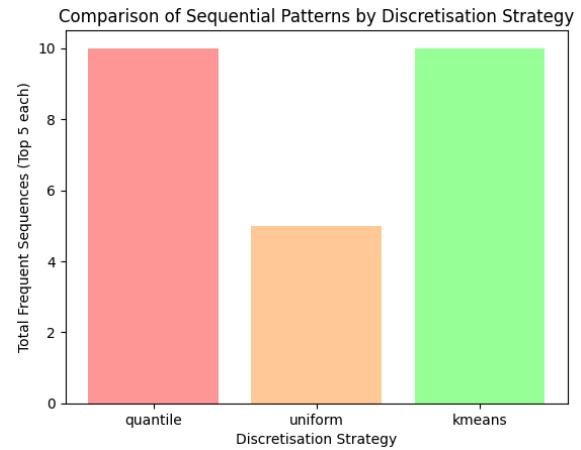
```

==== Strategy: QUANTILE ====
Top Malignant Patterns:
['radius_mean', 'perimeter_mean'] (support=167)
['radius_mean', 'perimeter_mean'] (support=166)
['radius_mean', 'perimeter_mean', 'area_mean'] (support=165)
['radius_mean', 'perimeter_mean', 'area_mean', 'smoothness_mean'] (support=78)
['radius_mean', 'perimeter_mean', 'area_mean', 'smoothness_mean', 'compactness_mean'] (support=36)
Top Benign Patterns:
['smoothness_se'] (support=97)
['smoothness_se', 'smoothness_worst'] (support=23)
['smoothness_se', 'symmetry_se'] (support=36)
['smoothness_worst'] (support=41)
['smoothness_mean'] (support=82)

==== Strategy: UNIFORM ====
Top Malignant Patterns:
['concave_points_worst'] (support=69)
['smoothness_worst'] (support=21)
['radius_mean'] (support=22)
['radius_mean', 'perimeter_mean'] (support=20)
['perimeter_mean'] (support=22)
Top Benign Patterns:

==== Strategy: KMEANS ====
Top Malignant Patterns:
['radius_mean'] (support=107)
['radius_mean', 'perimeter_mean'] (support=99)
['radius_mean', 'perimeter_mean', 'smoothness_mean'] (support=23)
['radius_mean', 'perimeter_mean', 'compactness_mean'] (support=27)
['radius_mean', 'perimeter_mean', 'concavity_mean'] (support=26)
Top Benign Patterns:
['texture_se'] (support=28)
['texture_worst'] (support=39)
['fractal_dimension_mean'] (support=32)
['smoothness_mean'] (support=33)
['smoothness_se'] (support=27)

```



Across the 3 discretisation strategy, we can see that quantile and kmeans produce similar robust patterns for malignant tumours, with size-related features consistently dominating 160 patients. Quantile binning clearly separates the sized-related features from smoothness-related features, while kmeans offers similar key patterns and better handles outliers. In contrast, uniform binning yields a weaker pattern with moderate support of 20-70 patients. This is likely due to unevenly distributed data causing extreme values to dominate bins. Overall, kmeans and quantile show the most reliable and consistent sequential patterns.

3.0 Task 3: EHR Data Synthesis with Machine Learning

3.1 Abstract

The growing availability of huge medical data sets has transformed the healthcare sector by enabling data-driven decision-making. Yet, patient privacy challenges, including stringent legislations and difficulty in obtaining good quality medical data has been extremely daunting. The generation of synthetic data has appeared as a potential solution that can solve these challenges. This study examines the performance of four distinct generative models that are designed for generating synthetic medical data: Tabular Variational Autoencoders (TVAE), Conditional Tabular Generative Adversarial Networks (CTGAN), and Gaussian Copula Variational Autoencoders (GCVAE). The performance of these models is evaluated by examining their capacity to successfully replicate the statistical characteristics of the medical dataset of Wisconsin Breast Cancer dataset. The comparison is centred around the usefulness of the created synthetic data for training machine learning models and its capability in maintaining statistical properties of the original data. The findings demonstrate that, despite each model possessing individual strengths, TVAE and GCVAE are more successful in maintaining statistical properties and creating valid synthetic data for healthcare uses.

3.2 Objective

This project aims to compare the performance of three synthetic data generative models for synthetic medical data generation:

Tabular Variational Autoencoders [2] (TVAE), uses a variational autoencoder-based neural network technique to train a model and generate synthetic data.

Conditional Tabular Generative Adversarial Networks [3] (CTGAN), a deep learning synthetic data generation method for single table data based on GANs that can learn from real data and produce highly accurate synthetic data.

Gaussian Copula Variational Autoencoders [4] (GCVAE), a variant of CopulaVAE that specifically uses Gaussian copulas to model feature dependencies.

From these models, this project aims to assess and propose the best model out of these three based on the model's ability to capture data utility and data similarity by assessing their Mean Absolute Error (MSE), Coefficient of Determination (R2), Mean squared error (MSE), Pearson Correlation, Wasserstein Distance, Column Shape Score, Column Pair Trends, Accuracy and AUC.

3.3 Background

3.3.1 Tabular Variational Autoencoder (TVAE)

Tabular Variational Autoencoders [2] (TVAE) represent a model type created to produce synthetic data for tabular datasets featuring continuous and categorical information. The standard Variational Autoencoder (VAE) architecture gets modified when applied to tabular data through TVAE. Identification of input data occurs through an encoder-decoder structure in standard VAE implementation where the encoder derives latent space information from inputs and the decoder produces data from this latent representation. TVAE develops the VAE model by designing separate distribution models for categorical and continuous features while discovering mutual relationships between them. The architecture of TVAE enables efficient processing of data collections like medical information because it effectively handles the intricate pattern of continuous and categorical variables.

The TVAE system successfully handles various data problems which occur specifically in medical databases. Medical datasets commonly present an imbalance issue that reduces the frequency of specific conditions. TVAE masterfully combines latent space representations, so it efficiently analyses both regular and uncommon health conditions. The features in medical data create elaborate interconnected patterns because medical history has well-defined relationships with current health indicators. TVAE effectively models dependencies in the data which produces realistic synthetic data that maintains real data statistical characteristics.

3.3.2 Conditional Tabular Generative Adversarial Networks (CTGAN)

The framework of GAN serves as the basis for Conditional Tabular Generative

Adversarial Networks [3] (CTGAN) model. CTGAN contains two systems that operate as the generator responsible for developing synthetic information while the discriminator verifies real versus synthetic data. A conditional generator serves as part of CTGAN by learning to generate synthetic data from features which specify certain input conditions. By implementing this mechanism, the model demonstrates its capability to analyse intricate relationships which exist between categorical and continuous variables in tabular data.

CTGAN demonstrates effective performance on medical datasets because these datasets tend to display class imbalance issues. The conditional generator of CTGAN enables synthetic data generation that reflects both total data distributions and hard-to-find dataset instances which appear infrequently in original data. CTGAN efficiently constructs synthetic patient records through its modelling of connections between categorical alongside continuous features which benefits healthcare applications that require complete patient records modelling.

3.3.3 Gaussian Copula Variational Autoencoder (GCVAE)

Gaussian Copula Variational Autoencoder [4] (GCVAE) functions as an extension of CopulaVAE to specifically implement Gaussian copulas for identifying feature dependency patterns. The continuous data modelling with Gaussian copulas depends on a joint normal distribution assumption for the features. GCVAE provides the advantages of variational inference and Gaussian copulas to effectively produce synthetic data that keeps the exact feature marginal distributions and preserves dependencies between features.

The medical nature of continuous variables makes GCVAE an exceptional approach to handle such datasets. The distribution characteristics of blood pressure, cholesterol levels, and heart rate resemble Gaussian distributions, which makes Gaussian copulas optimal for dependency modelling in this category of medical features. The data synthesis capabilities of GCVAE retain all interrelations between patient characteristics for valuable applications such as tailored medicine services, disease evolution modelling, and healthcare prognostic systems.

3.4 Data Preparation

3.4.1 Dataset

The dataset that we have chosen for this experiment is the Wisconsin Breast Cancer dataset as it represents a large-scale medical dataset that is well documented. Wisconsin Breast Cancer Dataset [5] is one of the most common datasets with features extracted from digitized images of FNA of breast mass of breast cancer tumour patients. It includes both continuous variables (e.g., tumour size, texture, etc.) and categorical labels (e.g., benign or malignant diagnosis) which will act as target labels for our classification models. This makes it an ideal candidate for testing synthetic data generation models for medical diagnosis.

3.5 Results

	MAE	R ²	MSE	Pearson Correlation	WD	Accuracy	AUC Score
CTGAN	0.459824	0.01047	0.231317	0.197499	13.357686	0.618629	0.636
TVAE	0.088893	0.781115	0.051168	0.888751	4.467438	0.933216	0.98141
GCVAE	0.245325	0.603092	0.092783	0.805961	3.761204	0.898067	0.959701

Table 3-1: Results of Synthetic Data Generation models

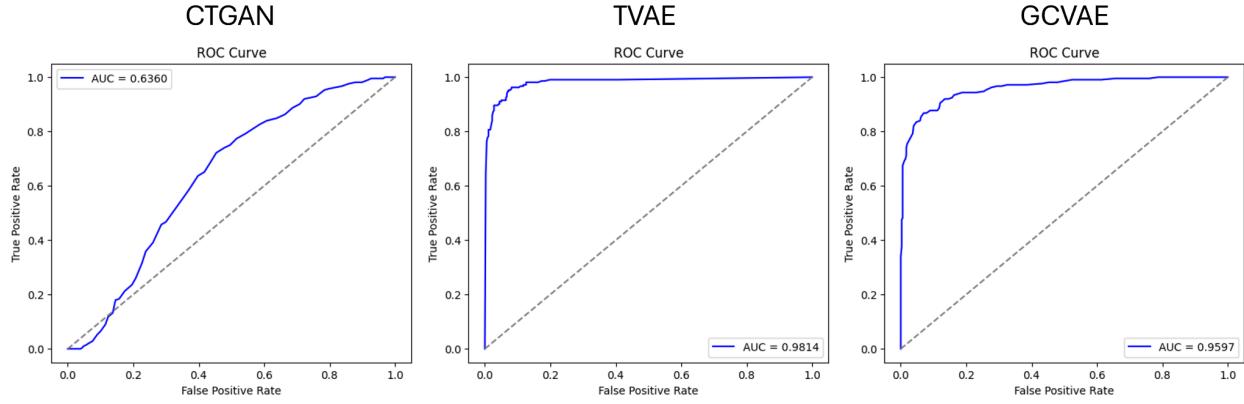


Figure 3-1: ROC Curve for CTGAN, TVAE, GCVAE

For synthetic data generated, TVAE has shown the lowest MAE, MSE. It also performed the best in terms of R2, Pearson Correlation, accuracy and AUC score. This shows that the TVAE model was able to produce the lowest reconstruction errors and the highest correlation and predictive fidelity. TVAE was able to produce and capture the complex joint dependencies in the electronic health dataset in the breast cancer dataset while generating data that remains highly useful for classifications.

GCVAE was also trailing behind as close second in above scenarios and managed to outperform TVAE in Wasserstein Distance. This may indicate a better global distribution alignment despite its predictive utility being lower than TVAE's.

CTGAN performed the worst out of the three despite its ability to model mixed data types as it suffered from higher reconstruction error and weaker correlation. This indicates that the model may not be a good fit for this small biomedical dataset.

Overall, TVAE offered the best results out of the three models as it was able to synthesise tabular medical records that retain statistical realism without reproducing the exact same patient sample as the original dataset. Future work for this project could include differential privacy regulation or including models that can incorporate temporal data such as time-series signals and event logs and unstructured data such as clinical notes. These options could be explored by including models such as PATE-GAN [6] or DP-GAN [7] and through models such as Time-GAN [8] or text-based GAN variants that generate richer, multimodal synthetic data.

References

- [1] D. Mwiti, 'Apriori Algorithm Explained: A Step-by-Step Guide with Python Implementation', datacamp. Accessed: Apr. 15, 2025. [Online]. Available: <https://www.datacamp.com/tutorial/apriori-algorithm>
- [2] H. Ishfaq, A. Hoogi, and D. Rubin, 'TVAE: Triplet-Based Variational Autoencoder using Metric Learning', Feb. 08, 2023, arXiv: arXiv:1802.04403. doi: 10.48550/arXiv.1802.04403.
- [3] L. Xu, M. Skoulikidou, A. Cuesta-Infante, and K. Veeramachaneni, 'Modeling Tabular data using Conditional GAN', Oct. 28, 2019, arXiv: arXiv:1907.00503. doi: .48550/arXiv.1907.00503.
- [4] D. P. dos Santos and J. C. S. Vasconcelos, 'Using Gaussian Copulas and Generative Adversarial Networks for Generating Synthetic Data in Beet Productivity Analysis', Sugar Tech, vol. 27, no. 2, pp. 407–417, Apr. 2025, doi: 10.1007/s12355-024-01506-w.
- [5] N. Patki, R. Wedge, and K. Veeramachaneni, 'The Synthetic Data Vault', in 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA), Montreal, QC, Canada: IEEE, Oct. 2016, pp. 399–410. doi: 10.1109/DSAA.2016.49.
- [6] O. M. William Wolberg, 'Breast Cancer Wisconsin (Diagnostic)'. UCI Machine Learning Repository, 1993. doi: 10.24432/C5DW2B.
- [7] J. Jordon, J. Yoon, and M. van der Schaar, 'PATE-GAN: Generating Synthetic Data with Differential Privacy Guarantees', presented at the International Conference on Learning Representations, Sep. 2018. Accessed: Apr. 15, 2025. [Online]. Available: <https://openreview.net/forum?id=S1zk9iRqF7>
- [14] R. Torkzadehmahani, P. Kairouz, and B. Paten, 'DP-CGAN: Differentially Private Synthetic Data and Label Generation', Jan. 27, 2020, arXiv: arXiv:2001.09700. doi: 10.48550/arXiv.2001.09700.
- [8] J. Yoon, D. Jarrett, and M. van der Schaar, 'Time-series Generative Adversarial Networks', in Advances in Neural Information Processing Systems, Curran Associates, Inc., 2019. Accessed: Apr. 15, 2025. [Online]. Available: https://papers.nips.cc/paper_files/paper/2019/hash/c9efe5f26cd17ba6216bbe2a7d26d490-Abstract.html