Synthetic generation of multi-modal discrete time series using transformers

Terrence Tricco
Memorial University

MEMORIAL UNIVERSITY

Kyle Nickerson, Antonina Kolokolova (Memorial), Ting Hu (Queen's), John Hawkin, Charles Robertson, Farzaneh Shoeleh (Verafin)

Synthetic Data

Synthetic (adj) – Prepared or made artificially.

Synthetic Data

Synthetic data (noun) – Artificially created data.

Synthetic Data

Synthetic data (noun) – Artificially created data that has the same properties as real data.

Synthetic Data

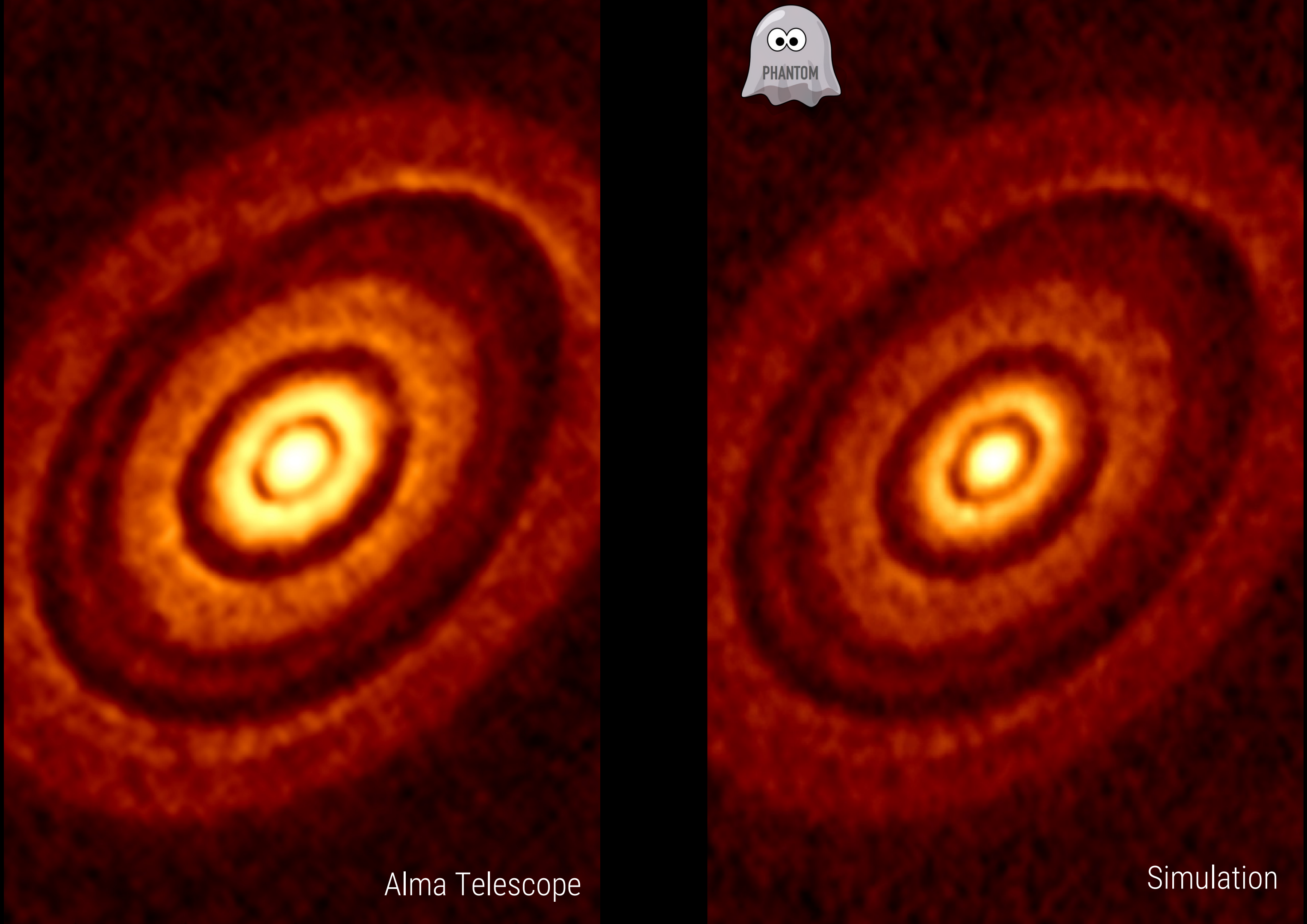
- What constitutes high-quality synthetic data?
  - Human judgment – is it believable?
  - Statistical properties similar to the real data.
  - Non-linear correlations between features are maintained.

Benefits of Synthetic Data

- Synthetic data can offer many benefits.
  - On demand production of as much data as needed.
  - Generation of rare events or labels.
  - Full knowledge of all data features and labels.
  - Sharing of data – avoiding potential privacy, legal or security concerns.

How to Create Synthetic Data

- Many scientific fields have used modeling and simulation for decades to create synthetic data.


PHANTOM
Alma Telescope
Simulation

Rise of the Machines (Learning)

- Simulation typically requires crafting the underlying processes by hand.

- **Generative machine learning techniques** can learn these processes without explicitly hard-coding them.
- Many techniques available nowadays, for example,
  - Generative Adversarial Networks (GANs),
  - Variational Auto-Encoders (VAEs),
  - Transformers, etc.

Synthetic Financial Transaction Data

- We are building a robust synthetic data generator for **personal financial transaction histories.**

- Research is in collaboration with Verafin (Hawkin, Robertson), a Nasdaq-owned software company specialized in preventing financial crime.

Synthetic Multi-Modal Discrete Time Series

- Our synthetic data is discrete time-series data that has:
  - Multivariate sequences, with multiple classes of events,
  - Events which are non-uniformly spaced in time,
  - Multi-modal patterns of activity on varying timescales,
  - With dependence upon a variety of metafactors (age, gender, etc).

Synthetic Multi-Modal Discrete Time Series

- Need to capture the relationships **between features** for each time event,
- And the complex relationships between events **across time**.

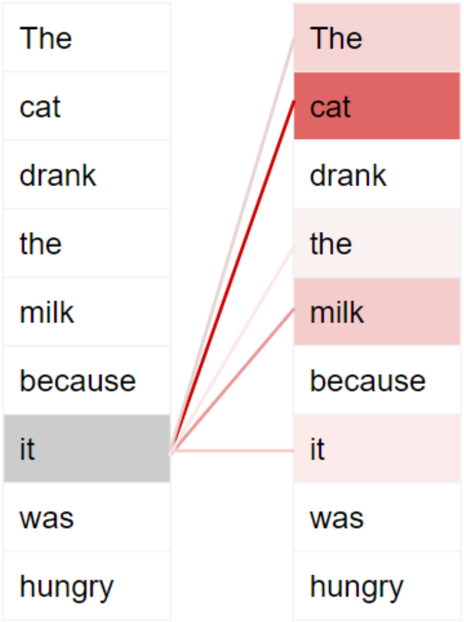
- Generative machine learning techniques promises a way to capture non-linear behaviours without needing to explicitly model these by hand.

Transformers

- Our approach uses an architecture based on **transformers**.
- Transformers have found great success in modelling sequences – especially for natural language tasks (e.g. Google's BERT, OpenAI GPT-3).

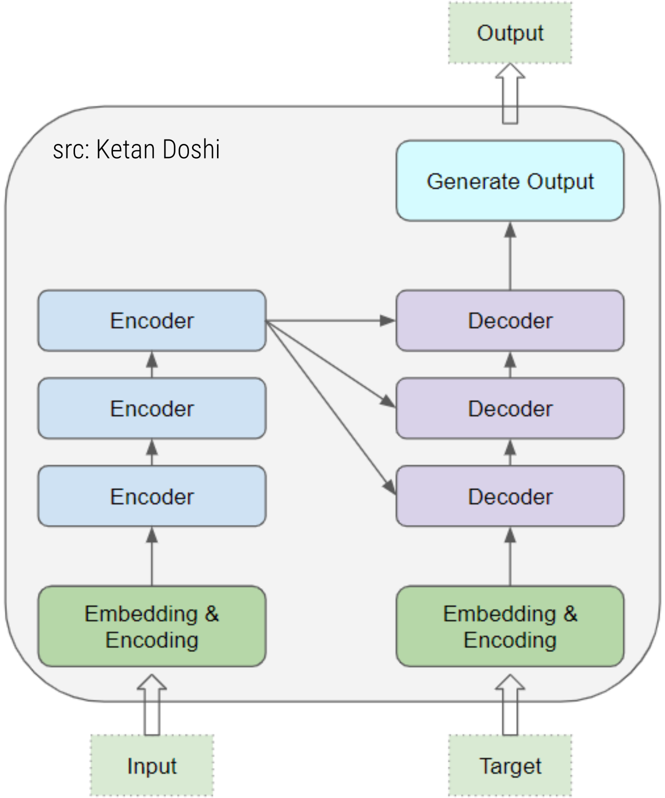
Transformers

- Transformers rely on a **self-attention** mechanism.
- Attention assigns weight individually between all elements in a sequence.
- This is different than recursion methods (e.g. RNN), where element n can only be processed after processing all previous n-1 elements.



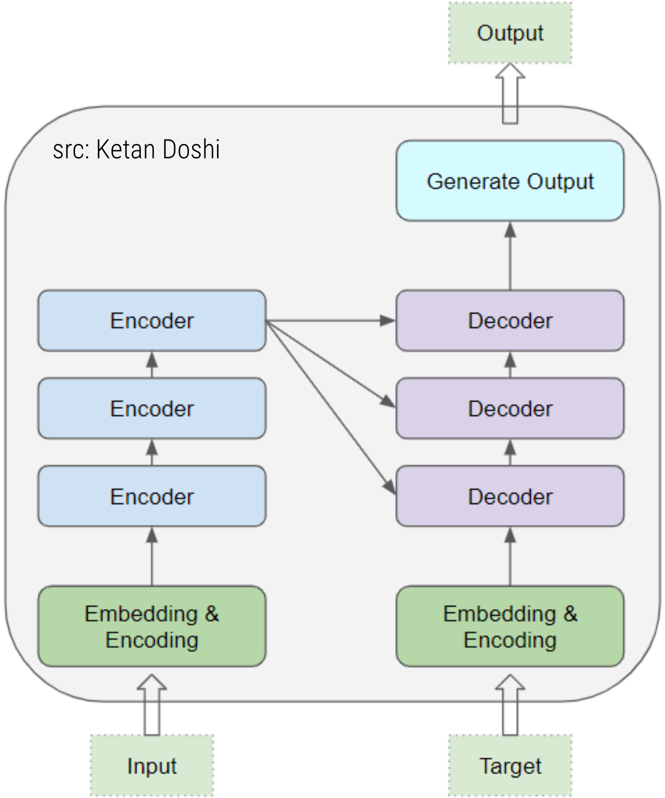
Transformers

- Transformers use an **encoder - decoder** framework.
- Input sequences are encoded using self-attention and positional encoding.
- Output sequences are decoded using the input encoding, plus self-attention between itself and the input/output.



Transformers

- Encoder layers include self-attention of the input sequence (how important each word is to all other words).
- Decoder layers include self-attention of itself + the encoded attention.
- The decoder stack is used to generate new sequences.



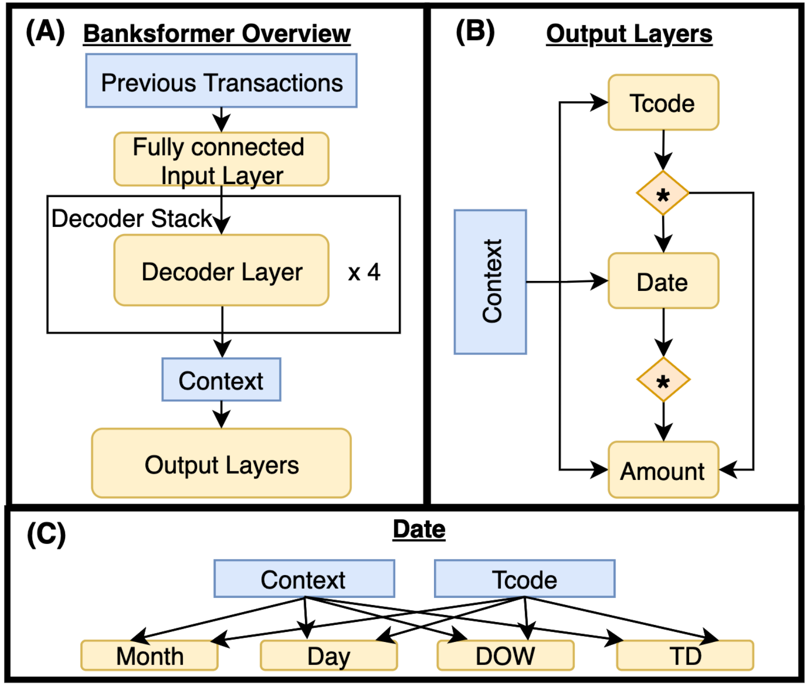
Banksformer

- **Banksformer** is our transformer-based model to generate a sequence of personal financial transaction sequences.

- For each transaction in the sequence, we generate the:
  - date/time of the next transaction,
  - type of transaction, and
  - value of the transaction.
- Customer age is included as influencing metadata.

Banksformer

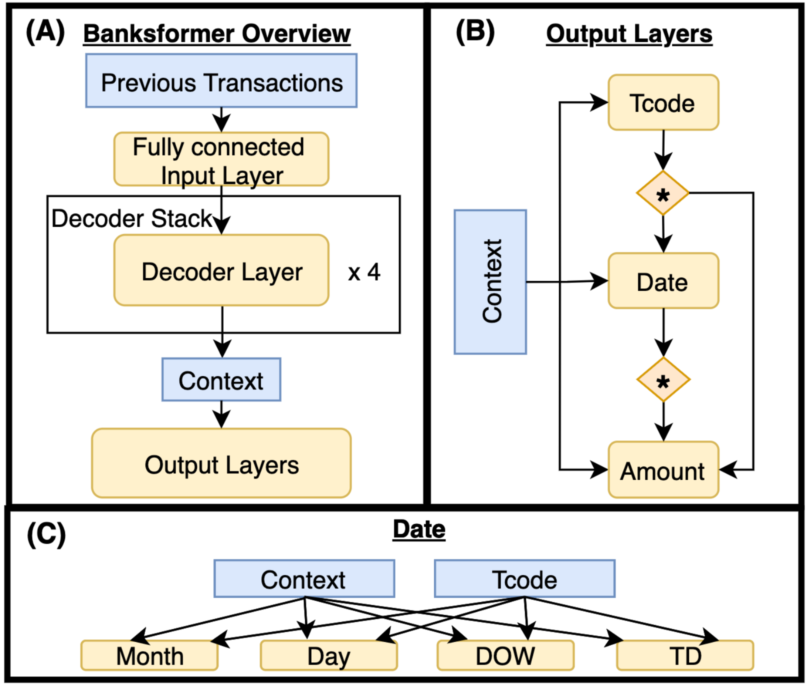
- We use the transformer-decoder architecture.
- Attention scores are calculated using scaled dot product.
- Loss function is the weighted sum of individual losses.
- Mean squared error is used for continuous features and categorical cross-entropy for categorical features.



Banksformer

- **Conditional generation** is used to capture **date-based patterns**.

- First, the transaction type is generated.
- Second, the date and time.
- Third, the dollar value of the transaction.



Date Encoding

- **Periodic encodings** are used to capture the periodicity in the date features.
- The month, day of the month, days till month's end, and day of week are each encoded as two features:

$$f_1 = \sin\left(2\pi i / n_i\right) \qquad f_2 = \cos\left(2\pi i / n_i\right)$$

- where *i* and n_i refer to the 7 days of the week, 12 months of the year etc.

- Data is generated by sampling from the joint probability distribution created from probability distributions for each individual feature.

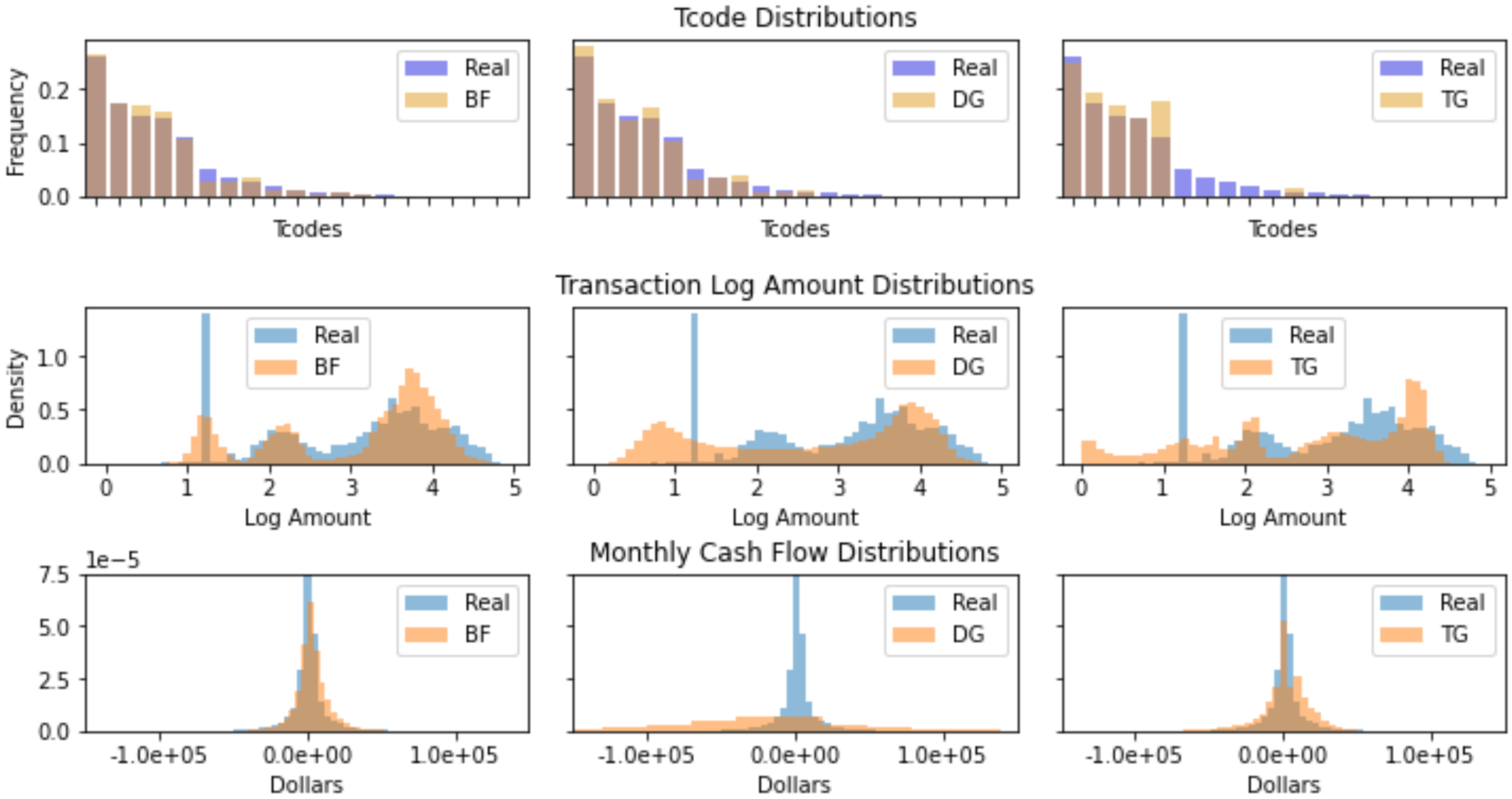
Baseline Comparison

- We compare to two GAN based models:
  - DoppleGANger
  - TimeGAN
- Both have been successful in generating time-series data.
- Include a number of adjustments specific for time-series data, for example, around their embeddings, loss functions, conditional generation, etc.

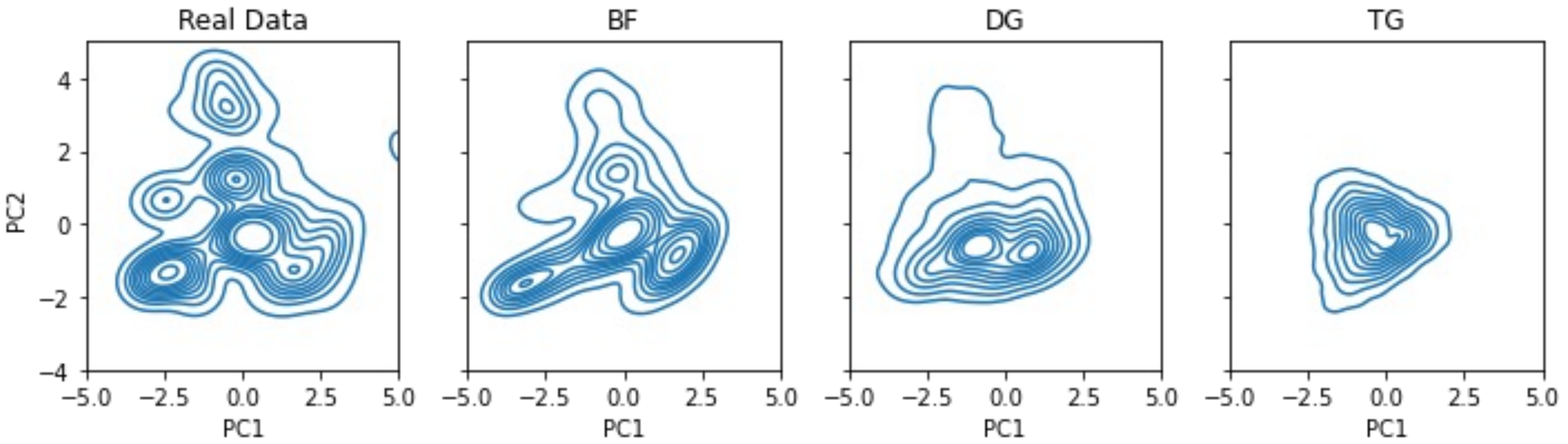
Data

- We use public, open-source data as a proof of concept.
- ~1M transactions over 4500 accounts from real, personal banking transactions in the Czech Republic over a 5-year period.
- ~100k transactions over 5000 accounts from synthetic banking transactions in the United Kingdom over a 2-month period.

Results – Univariate Distributions

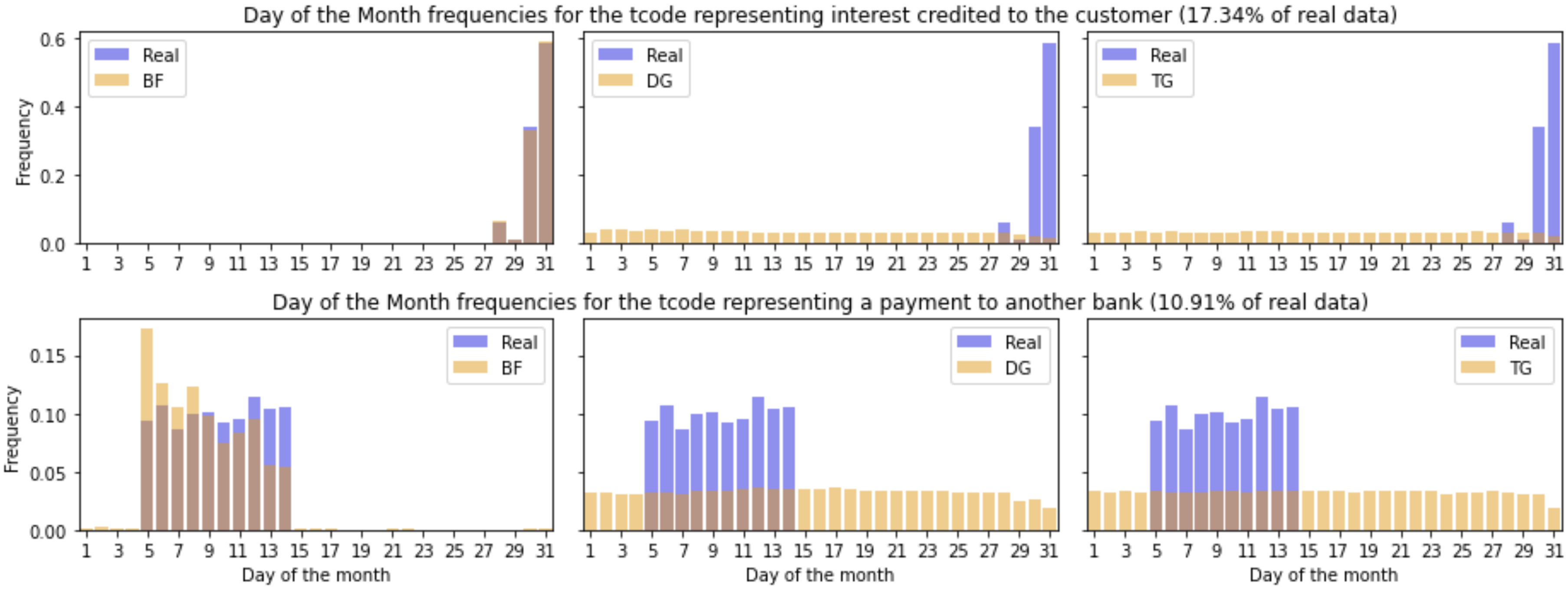


Results – Joint Distributions



- Sequences are flattened and a PCA model fit to the data.
- First two principal components are plotted.

Results – Date / Transaction Codes



- Two transaction types that are highly dependent upon day of month.
- Represents ~28% of all transactions.

Summary

- We have built a generative model for multi-modal discrete time-series sequences using **transformers**.
- Compared to two popular GAN models, we have found that our transformer model better represents **date/time patterns** and **joint distributions**.
- Key features are to encode the periodicity in date/time features, and use conditional generation to sequentially generate each feature.
- Future steps: realistic demographic data, more feature rich transactions (e.g. cheque images).