Towards a Foundation Purchasing Model: Pretrained Generative Autoregression on Transaction Sequences

Piotr Skalski Innovation Lab Featurespace Cambridge, UK piotr.skalski@featurespace.co.uk David Sutton
Innovation Lab
Featurespace
Cambridge, UK
david.sutton@featurespace.co.uk

Stuart Burrell Innovation Lab Featurespace Cambridge, UK

stuart.burrell@featurespace.co.uk

Iker Perez Innovation Lab Featurespace Cambridge, UK iker.perez@featurespace.co.uk Jason Wong Innovation Lab Featurespace Cambridge, UK

jason.wong@featurespace.co.uk

ABSTRACT

Machine learning models underpin many modern financial systems for use cases such as fraud detection and churn prediction. Most are based on supervised learning with hand-engineered features, which relies heavily on the availability of labelled data. Large selfsupervised generative models have shown tremendous success in natural language processing and computer vision, yet so far they haven't been adapted to multivariate time series of financial transactions. In this paper, we present a generative pretraining method that can be used to obtain contextualised embeddings of financial transactions. Benchmarks on public datasets demonstrate that it outperforms state-of-the-art self-supervised methods on a range of downstream tasks. We additionally perform large-scale pretraining of an embedding model using a corpus of data from 180 issuing banks containing 5.1 billion transactions and apply it to the card fraud detection problem on hold-out datasets. The embedding model significantly improves value detection rate at high precision thresholds and transfers well to out-of-domain distributions.

CCS CONCEPTS

• Applied computing \rightarrow Online banking; • Computing methodologies \rightarrow Unsupervised learning; Learning latent representations.

KEYWORDS

transaction embeddings, self-supervised learning, generative modelling, multivariate time series, fraud detection

1 INTRODUCTION

Foundation models have seen tremendous success and wide adoption within the past couple of years. They have proven their ability to leverage large corpora of data and scale to hundreds of billions of parameters. On textual data, these models can be used not only to generate human-level text but also to produce contextualised

embeddings of individual tokens, sentences, and even whole documents that can be fed as inputs to downstream models. Their rapid success has been in no small part due to the development of self-supervised learning (SSL) methods such as autoregressive [27] and masked [13] language modelling which have allowed models to learn contextual representations of input tokens without relying on labels.

While these methods have already been successfully used with different modalities such as natural language [4, 11, 22, 27, 28], computer vision [26, 30], audio [3, 12], and tabular data [1, 20, 31] there has been little work to adapt them to the case of multivariate time series data. One example of such data modality of particular interest in this work is streams of financial transactions – sequences of events representing transfers of funds between two entities. Each event can be described by a set of numerical or categorical features, such as the timestamp, card number, transaction amount, merchant name, or merchant category (in the case of card transactions).

From the perspective of financial institutions, the most important modelling problems in this domain include fraud detection, money laundering detection, credit default prediction, customer churn prediction, and future expenditure modelling. Most common approaches to solve these problems are based on supervised learning and rely on hand-engineered features which take time and domain expertise to define for specific modelling problems. These approaches are therefore not amenable to transfer learning and require redesigning of feature definitions when new fraud typologies emerge.

Self-supervised learning has the potential to replace the expensive feature engineering process in favour of learnt representations from a foundation model pretrained on large quantities of unlabelled data. However, efforts in this space have so far been limited. Recently, a contrastive learning SSL approach was designed to generate embeddings of cardholders based on their transaction history [2]. These embeddings were evaluated on entity-level classification tasks and shown to perform on par and in some cases better than hand-engineered features. However, autoregressive language modelling approaches have not yet been adapted to the domain of financial transactions, even though the task of predicting future events bears a close resemblance to modelling problems in the financial industry.

^{2023.} This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in 4th ACM International Conference on AI in Finance (ICAIF '23), November 27–29, 2023, Brooklyn, NY, USA, https://doi.org/10.1145/3604237.3626850.

Skalski, et al.

The success of generative pretraining methods in NLP and lack of equivalent approaches in the domain of financial transactions has motivated our research. In this paper, we present a self-supervised learning method for pretraining autoregressive models that can generate transaction embeddings. Our pretraining method NPPR combines two objectives: next event prediction (NP) and past reconstruction (PR). The next event prediction task, adapted from language modelling to handle multivariate transaction events, was motivated by the similarity between generative modelling and the financial modelling tasks such as churn, credit default and expenditure prediction. All of these tasks aim to predict the future actions by an entity, and solving them requires the model to encode features capturing behavioral characteristics of entities. The past reconstruction task serves the purpose of further encouraging the model to learn longer-term behavioral features which increase the predictive performance of the embeddings on downstream problems.

We evaluate our method on four publicly available datasets of card transactions, showing that the generated embeddings can outperform hand-engineered features and other SSL methods on churn prediction, age group classification, expenditure forecasting, and credit default prediction. We furthermore use our method to pretrain a Foundation Purchasing Model on a large corpus of transaction histories from 180 European issuing banks and use the model to produce transaction embeddings on three hold-out issuer datasets that were excluded from the pretraining corpus. The hold-out issuers operate in a different country to any of the pretraining issuers. We apply these embeddings to the fraud detection problem, showing transferability of the model to significantly out-of-domain data and benefits of pretraining on a large and diversified corpus of transactions. Visualisations of the embedding space show that the model encodes similarity among different merchant category codes akin to semantic similarity of word embeddings learnt by large language models.

To summarise, in this paper we make the following contributions:

- propose a self-supervised learning method that combines a next event prediction task with a past reconstruction task, both adapted to the domain of multivariate time series of financial transactions;
- (2) show that our method outperforms hand-engineered features and other pretraining methods on downstream classification and regression tasks using evaluations on public datasets;
- (3) demonstrate that pretraining with our method on a large corpus of card transaction datasets from 180 issuing banks improves fraud detection at high precision thresholds and transfers well to out-of-domain data;
- (4) illustrate that the resulting embeddings are able to capture semantic similarity between merchant category codes.

2 RELATED WORK

Many SSL tasks for sequential data were originally designed for the domain of natural language. Autoregressive language modelling aims to predict the next token in a sentence based on the previous ones and has been used to pretrain the GPT family of models [4, 27, 28]. In masked language modelling (MLM) [13], randomly sampled tokens are masked with a special mask token and the

network is tasked with predicting the original token. This method has been successfully adapted to other domains including vision [26, 30], audio [3, 12], and tabular data [1, 20]. Next sentence prediction [13] has been used together with MLM and works by feeding the network two sentences A and B and predicting whether B follows A. Replaced token detection, used by ELECTRA [11], is a modification of MLM where randomly sampled tokens are replaced with candidates generated by a different language model and the task is to predict the original input token.

Another popular class of SSL methods is contrastive learning. Typically, it learns representations that are invariant to data augmentation. It involves generating positive and negative pairs where the positive pairs come from two augmented views of the same sample, while negative pairs come from two different samples. A contrastive loss function encourages representations to be similar for positive pairs and dissimilar for negative pairs. Some examples include SimCLR [8] for images (which uses a composition of standard image augmentation methods), SAINT [29] for tabular data (uses CutMix [35] in input space and mixup [37] in latent space), SimCSE [15] (applies dropout as data augmentation). CPC [34] is a variation of contrastive learning applicable to autoregressive models that tries to maximize the mutual information between the hidden state and future events from the same sequence. In the domain of financial transactions, to the best of our knowledge, CoLES [2] is the only method that has used contrastive learning to obtain entity embeddings. It uses randomly generated subsequences from a transaction history belonging to the same entity as positive pairs and subsequences from different entity histories as negative pairs.

There also exist non-contrastive methods which train representations invariant to data augmentation using positive examples only. They avoid representation collapse by using a momentum encoder (BYOL [18], TiCo [38]), penalizing cross-correlation between positive views (Barlow Twins [36]), clustering embeddings with an equipartition constraint (SwAV [7]), and applying an asymmetrical stop-gradient operation (SimSiam [9]).

Financial transactions have also been used with graph-based methods where originators and beneficiaries (and sometimes transactions) form the nodes of a graph. Some early work in this area involve supervised training for fraud detection [17, 23, 25] and generating merchant embeddings [5, 16]. More recently, there has also been progress in inductive representation learning. Graph-Sage [19] encourages similar representations for nearby nodes and has been used to create embeddings for credit card fraud detection [32, 33]. Link prediction between nodes has been used as an SSL task for detecting anomalous transactions [6] and capturing inter-company relationships [24]. However, none of these methods leverages the inherent time-series nature of transactions. They are typically investigated in the context of money laundering detection, where patterns of movement of funds across a network are analysed periodically in a batch process. In contrast, in this work, we are interested in generating transaction embeddings for applications that require real-time processing, such as fraud prevention.

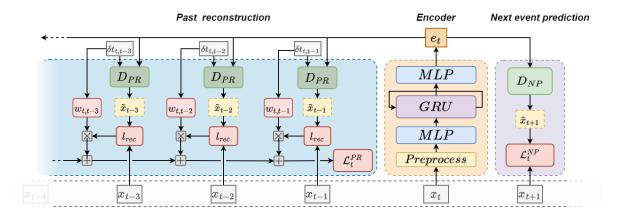


Figure 1: The NPPR generative modelling framework for pretraining a recurrent encoder using a combination of next event prediction and past reconstruction tasks.

3 GENERATIVE MODELLING ON TRANSACTION SEQUENCES

3.1 Proposed method

Suppose $\mathcal{H}=\{h_e\}$ is a set of transaction histories pertaining to some entities such as cardholders or account holders. A transaction history is a time-ordered sequence $h_e=\{x_t\}_{t=0}^{T_e}$ of financial transactions, where each transaction $x_t\in X$ is described by a set of numerical and categorical features (amount, merchant name, etc.). ¹ The goal is to train an encoder network $E\colon X^*\to \mathbb{R}^d$ that creates an embedding vector $e_t\in \mathbb{R}^d$ of a transaction x_t given past transactions from the same entity up to this one, i.e. $e_t=E(x_t,x_{t-1},...,x_0)$. To extend this setup to any multivariate time series data, we will refer to transaction histories as *sequences* and individual transactions as *events*. The proposed self-supervised algorithm is visualised in Figure 1 and is composed of two tasks.

Next event prediction (NP) is the primary task and adapts autoregressive language modelling to the case of multivariate events. A decoder network $D_{NP} \colon \mathbb{R}^d \to \mathcal{X}$ takes an embedding e_t of event x_t to generate predictions of the next event's features $\tilde{x}_{t+1} = D_{NP}(e_t)$. For numerical features the predictions are simply real numbers while for categorical features they are vectors of probabilities over the distinct categories of the feature. The objective function is defined as

$$\mathcal{L}_{t}^{NP} = \sum_{f} l_{rec}^{f} \left((\tilde{x}_{t+1})_{f}, (x_{t+1})_{f} \right) \tag{1}$$

where $()_f$ denotes a slice of a vector corresponding to a particular feature f and the reconstruction loss function l_{rec}^f for a single feature f is mean squared error if f is a numerical feature and cross-entropy if f is a categorical feature.

Past reconstruction (PR) is the secondary task that aims to guide the encoder towards learning behavioral features with long-term dependencies. We define a decoder network $D_{PR}: (\mathbb{R}^d, \mathbb{R}) \to \mathcal{X}$ that takes as input an embedding e_t of event x_t and a scalar time difference $\delta t_{t,t-k}$ between this event and an event from the

past x_{t-k} and generates a reconstruction of that past event $\tilde{x}_{t-k} = D_{PR}(e_t, \delta t_{t,t-k})$. The objective function is a weighted sum of reconstruction losses of past events:

$$\mathcal{L}_{t}^{PR} = \sum_{k=1}^{\min(K,t)} \omega_{t,t-k} \sum_{f} l_{rec}^{f} \left((\tilde{x}_{t-k})_f, (x_{t-k})_f \right)$$
 (2)

where $\omega_{t,t-k} = exp(-\delta t_{t,t-k}/\lambda)$ is a weight function that decays exponentially with the time difference between events at a rate governed by the decay length hyperparameter λ . The summation over the past events is truncated so that at most K past events contribute to the loss.

For a sequence of events h_e , we define the total loss as the sum of event losses that are weighted combinations of the two objective functions defined above:

$$\mathcal{L}_{e} = \sum_{t}^{T_{e}} (1 - \alpha) \mathcal{L}_{t}^{NP} + \alpha \mathcal{L}_{t}^{PR}$$
 (3)

where $\alpha \in (0, 1)$ is a hyperparameter.

3.2 Model architecture

Although any autoregressive model architecture can be used as an encoder, we decided to use a recurrent model based on GRUs [10]. Compared to unidirectional transformer models, RNNs are more efficient in production where new events arrive one at a time since they only have to store and process hidden state and a new event rather than a whole sequence of previous events. This consideration is often very important in the financial sector when using a model for real-time decisioning, where there are often stringent requirements on response latency. The encoder architecture is shown in Figure 1. An event x_t is first preprocessed into a dense vector. Numerical features are normalized while categorical features are encoded through an embedding layer. Additionally, event timestamps are used to produce a numerical feature that encodes the time gap between events x_t and x_{t-1} belonging to the same entity. The resulting vectors for individual features are concatenated to form a single dense representation of an event. This vector is then passed through a stack of layers $\phi_{proj} \circ \phi_{GRU} \circ \phi_{MLP}$ that enriches

 $^{^{1}}$ For clarity of notation the subscript e on transactions has been dropped.

Skalski, et al.

it with the representation of past events. The use of MLP before the GRU layer adds expressivity to the encoder and the projection layer ϕ_{proj} allows the size of the hidden state to scale independently of the final embedding size.

The decoder models in the two modelling heads are simple MLPs. The output of the last dense layer (with linear activation) in each decoder is split into multiple vectors, one for each feature in the encoded events. The vectors corresponding to numerical features have size one (scalar), while those corresponding to categorical features have size equal to the number of distinct values the feature can take. The vectors for categorical features are additionally passed through a *softmax* activation to produce probability estimates.

4 EXPERIMENTS ON PUBLIC DATASETS

In this section we evaluate the performance of entity-level embeddings generated with our method when used in classification and regression tasks. We use the embeddings as inputs to downstream models without fine-tuning the pretrained models. This evaluation setup is suitable for testing systems where the embeddings server and downstream modelling setup are decoupled. Code to reproduce experiments in this section is publicly available on GitHub². Since publicly available transaction datasets are very limited and their schemas are often incompatible, pretraining and evaluation was performed separately on each dataset. Pretraining on a large corpus of data and out-of-domain evaluation on hold-out datasets is investigated in the next section using private datasets.

4.1 Datasets

We use publicly available datasets from various data science competitions comprising debit and credit card financial transactions. These datasets include unlabelled and labelled transactions for four different tasks: age group prediction³, churn prediction⁴, future expenditure (expnd.) forecasting⁵, and credit default prediction⁶. Important statistics of each dataset are shown in Table 1. For the expenditure forecasting task, we split the original 4-month training period into a 3-month training period and a 1-month labelling period. We then construct one labelled example per entity by taking its transactions over the 3-month training period as the input and using its total expenditure over the 1-month labelling period as the label.

Training and test sets for each dataset were generated by using 80% of entity histories as the training set and the remaining 20% as the test set. The pretraining set was constructed by concatenating the unlabelled entity histories and the training set.

4.2 Hyperparameters

For pretraining, we used the same encoder architecture on each dataset: MLP with 2 hidden layers (512 neurons each) and ReLU activation followed by a GRU with hidden state size 512, followed by a dense projection layer (with sigmoid activation) to the embedding space of size 512. Both the NP and PR decoders in our method are MLPs with 2 hidden layers (512 neurons each) and ReLU activations.

Table 1: Characteristics of the four publicly available datasets of financial transactions.

labelled cards	unlabelled cards	num. features	cat. features
5K	5K	1	4
30K	20K	1	1
400K	0	2	6
960K	510K	1	14
	5K 30K 400K	cards cards 5K 5K 30K 20K 400K 0	cards cards features 5K 5K 1 30K 20K 1 400K 0 2

For training, we used the Adam optimiser with learning rate 10^{-3} and early stopping.

For pretraining with our method, the decay length λ was chosen to be 2 months based on domain expertise. The hyperparameter α controlling the proportion of past reconstruction task in the total loss was tuned on each dataset separately using cross-validation: 0.1 (churn), 0.001 (age), 0.005 (expenditure), and 0.001 (credit default). For downstream model training, we used MLPs with 3 hidden layers (512 neurons each on churn and age prediction, and 1024 neurons each on expenditure and credit default prediction) and ReLU activations together with dropout and weight decay regularisation. Dropout rate, weight decay, and learning rate were tuned on each baseline separately using the Optuna framework with 5-fold cross-validation.

4.3 Baselines

We compare our approach against four baselines.

Hand-engineered features. We use the same hand-crafted features as in [2]. For numerical features, we apply aggregate functions (sum, count, mean, min, max, variance) over all transactions in an entity history. For categorical features, we compute the above aggregates of numerical features within groups of transactions grouped by every unique value of each categorical feature.

SimCSE. This uses sequence embeddings pretrained with contrastive learning where dropout was used as the data augmentation strategy for generating positive pairs. The dropout rate was tuned on each dataset.

Replaced event detection (RED). This is an adaptation of the *replaced token detection* task used in ELECTRA [11] where randomly sampled events are swapped for random events from other sequences in a batch, and the decoder is tasked with predicting the sampling mask. We found that a sampling probability of 30% performed best on downstream tasks.

CoLES. A contrastive learning method where positive samples are random subsequences coming from the same sequence, and negative samples are subsequences from two different sequences. This method is sensitive to the choice of minimum and maximum subsequence sampling lengths. We adopted the same values of these hyperparameters as in the original paper [2].

²https://github.com/Featurespace/foundation-model-paper

 $^{^3} https://ods.ai/competitions/sberbank-sirius-less on$

⁴https://boosters.pro/championship/rosbank1/

⁵https://ods.ai/competitions/x5-retailhero-uplift-modeling

⁶https://boosters.pro/championship/alfabattle2/overview

Table 2: Evaluation of self-supervised embeddings on downstream tasks. Average test set performance and standard deviation values from multiple runs on different training set folds.

Method	Churn	Age	Expnd.	Default
	AUC↑	Accuracy↑	MSLE↓	AUC↑
FeatEng SimCSE RTD CoLES NPPR	$\begin{array}{c} 0.798_{\pm 0.004} \\ 0.650_{\pm 0.006} \\ 0.827_{\pm 0.003} \\ 0.813_{\pm 0.003} \\ \textbf{0.845}_{\pm 0.003} \end{array}$	$\begin{array}{c} 0.626_{\pm 0.002} \\ 0.410_{\pm 0.002} \\ 0.590_{\pm 0.001} \\ 0.633_{\pm 0.002} \\ \textbf{0.642}_{\pm 0.001} \end{array}$	$\begin{array}{c} 0.743_{\pm 0.001} \\ 1.140_{\pm 0.001} \\ 0.747_{\pm 0.001} \\ 0.758_{\pm 0.001} \\ \textbf{0.723}_{\pm 0.001} \end{array}$	$\begin{array}{c} 0.768_{\pm 0.001} \\ 0.630_{\pm 0.001} \\ 0.765_{\pm 0.001} \\ 0.765_{\pm 0.001} \\ \textbf{0.798}_{\pm 0.001} \end{array}$

For each of the methods above, we used the embedding of the most recent event as the sequence embedding. For the NPPR method, we used the average of all event embeddings in a sequence as the sequence embedding, which can improve performance on downstream tasks as shown in the next section.

4.4 Results

Below we report results from each of the chosen methods using tuned hyperparameters. We report the mean and standard deviation on test sets from multiple training runs on different folds of the training set.

4.4.1 Comparison with baseline methods. Table 2 compares NPPR to the different baseline methods. Our method outperforms other methods on all datasets, including hand-engineered features, which turns out to be the strongest baseline (outperforming CoLES on two datasets, RTD on three datasets and SimCSE on all datasets). NPPR offers significant performance improvements on the churn, expenditure, and credit default prediction problems which aim to predict the future behavior of an entity. This confirms our hypothesis that generative modelling is particularly suitable for learning behavioral features that are predictive of future events. When predicting a static attribute of an entity, such as age group prediction, the contrastive CoLES method is competitive, but our method shows superior performance even in this case. This is achieved due to transaction embeddings averaging, which we demonstrate in section 4.4.3.

4.4.2 Importance of constituent tasks. Table 3 shows results of ablating the two constituent tasks from our method. In both cases, an entity embedding was constructed by averaging the transaction embeddings from the whole entity history as in the NPPR method. In general, embeddings pretrained with the next event prediction task perform significantly better than those using just past reconstruction task, except for churn prediction. In fact, using only the next event prediction task outperforms the other baselines from Table 2 on three out of four problems, which demonstrates the strength of vanilla generative modelling for learning behavioral features.

Even though the performance gap between the two tasks can be significant, as in the case of age group prediction, using a combination of both tasks outperforms pretraining with either of the two tasks in isolation on all datasets. Adding even a small amount of

Table 3: Ablation study comparing NPPR to next event prediction (NP) and past reconstruction (PR) tasks used in isolation. Average test set performance and standard deviation values from multiple runs on different training set folds.

Method	Churn AUC↑	Age Accuracy↑	Expnd. MSLE↓	Default AUC↑
NPPR	0.845 ±0.003	$0.642_{\pm 0.001}$	$0.723_{\pm 0.001}$	$0.798_{\pm 0.001}$
PR	$0.833_{\pm0.004}$	$0.542 _{\pm 0.002}$	$0.747_{\pm 0.001}$	$0.744_{\pm 0.001}$
NP	$0.814_{\pm0.002}$	$0.630_{\pm 0.002}$	$0.733_{\pm 0.001}$	$0.795_{\pm 0.001}$

Table 4: Relative performance difference on the test set between using averaged transaction embedding vs. embedding of a most recent transaction.

Method	Churn AUC↑	Age Accuracy↑	Expnd. MSLE↓	Default AUC↑
NPPR avg vs. last	-0.5%	+6.1%	-0.3%	+0.6%
CoLES avg vs. last	-1.0%	+0.3%	0.0%	-1.8%

past reconstruction loss to the total loss has a positive effect on the performance on all downstream problems. This suggests that the past reconstruction task encourages each transaction embedding to encode longer-term behavioral patterns which the next event prediction task doesn't explicitly do.

Interestingly, the past reconstruction task performed better than next event prediction on churn prediction. We hypothesize that reconstructing past events helps embeddings encode information from further back in time, which in turn allows the churn prediction model to more accurately model decline in transaction velocity. Consequently, the best performing value of α , which controls the contribution of the past reconstruction task to the total loss, was larger on churn prediction compared to other datasets.

4.4.3 Effect of averaging transaction embeddings. In this experiment, we evaluated the importance of using the average transaction embedding as an entity embedding by comparing it to the strategy adopted in the baseline methods, where the embedding of the most recent transaction was used instead. Table 4 shows the results of this evaluation.

We can see that embedding averaging can improve the performance of our method on downstream problems, especially in cases where the task involves predicting a static entity attribute such as age. However, it can also have a detrimental effect in problems such as churn prediction, presumably because averaging can oversmooth the features encoded in the more recent embeddings which capture a decline in the rate of transacting. By contrast, embedding averaging does not improve CoLES embeddings, which are designed to be similar across transactions by the same entity.

Skalski, et al

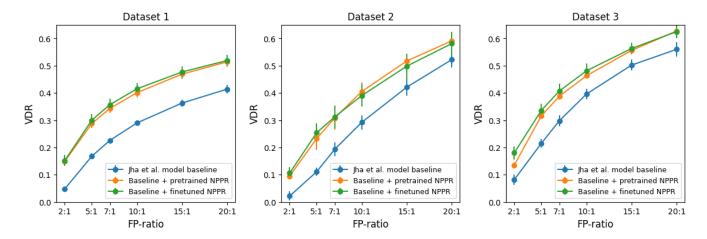


Figure 2: Evaluations of fraud detection models trained on datasets from three different issuers. Three models are shown: baseline hand-engineered features from Jha et al. [21], baseline features with NPPR embeddings trained on the pretraining corpus, and baseline features with NPPR embeddings finetuned on downstream datasets. Error bars were computed from multiple independent training runs.

5 APPLICATION TO FRAUD DETECTION AT SCALE

In this section, we apply our self-supervised NPPR method to pretrain a *Foundation Purchasing Model* on transaction data from a large number of issuing banks. We use it to produce transaction embeddings for unseen data, specifically transactions from hold-out issuing banks that operate in different countries to the issuers in the pretraining dataset. A separate fraud classifier is then trained on each of the hold-out issuers. We demonstrate that the pretrained model improves fraud detection performance, transfers well to significantly out-of-domain data, and learns semantic similarity between different merchant categories.

5.1 Pretrained embedding model

A single embedding model was pretrained on a corpus of card transaction datasets from 180 European issuing banks, each of which conforms to the ISO 8583 messaging format [14]. The corpus contains over 5.1 billion transactions which provide complete transaction histories covering a period of 12 months for 61 million cardholders.

The architecture of the embedding model is as described in the previous sections. It consists of an MLP with 2 hidden layers (2048 neurons each) with ReLU activations, followed by a GRU layer with state size 1024 and a final projection layer to an embedding space of size 768. The decay length λ in the NPPR method was 2 months, and the weight of past reconstruction task α was 0.001.

5.2 Fraud detection models

5.2.1 Datasets. For the downstream fraud detection task, we used labelled datasets from three European issuing banks. These datasets were not part of the pretraining corpus and correspond to issuers that operate in different countries to any of the issuers from the pretraining corpus. This presents the opportunity for testing transfer of the embedding model to significantly out-of-domain data.

The three datasets contain 11 months of transactions from 17 million, 3.5 million, and 1.8 million cardholders. We split each dataset into training, validation, and test sets both temporally and on the entity level, i.e. they contain transactions from different cardholders and non-overlapping consecutive time periods.

The fraud rate in each dataset is respectively 0.04%, 0.027%, and 0.11%. Due to the high class imbalance, we downsampled genuine transactions before training classification models (but no downsampling was performed for pretraining).

- 5.2.2 Baseline features. As a baseline, we used a representative traditional fraud prevention model drawn from literature [21]. It consists of primary transaction attributes and 14 hand-engineered behavioral features in the form of aggregations over windows of past transactions by the same entity at different time scales. Examples of such features include the average amount spent per transaction over the last month, or the total number of transactions with the same merchant during last month. We refer readers to the source paper for a detailed description of the features.
- 5.2.3 Models. Downstream classification models are MLPs with 3 hidden layers (1024 neurons each) and ReLU activations. They were trained with learning rate 10^{-3} , batch size 1024 and early stopping. Dropout was used as regularisation with rate 0.2.
- 5.2.4 Evaluation metrics. Production fraud prevention systems are often judged by their performance in reducing fraud losses while operating at high precision score thresholds. False positive predictions lead to declined transactions, which cause losses to the issuing bank and have a detrimental effect on consumer experience. An appropriate metric should measure the value of fraudulent transactions that have been prevented at a certain threshold of declined genuine transactions. A typical metric is the VDR @ FP-ratio where value detection rate (VDR) is the true positive rate weighed by transaction value and the threshold metric false positive ratio (FP-ratio) is the number of false positives divided by the number



Figure 3: t-SNE projection of a MCC embedding space. Each MCC embedding was obtained by averaging transaction embeddings corresponding to those MCCs.

of true positives. The classification score threshold is adjusted to reach a typical value of FP-ratio between 1:1 and 20:1.

5.3 Quantitative results

We evaluated three different classification models comparing baseline features, baseline features with NPPR embeddings from the pretrained model, and baseline features with NPPR embeddings from the pretrained model that has been finetuned on downstream datasets. Finetuning was performed in a self-supervised way using our NPPR method and is therefore applicable to the scenario where embeddings server and downstream modelling setup are decoupled. Results are shown in Figure 2 where VDR is plotted against different FP-ratio thresholds.

The addition of NPPR embeddings to the baseline features provides significant improvements in the value detection rate on all FP-ratio thresholds. At 5:1 FP-ratio our embeddings can provide up to 140% uplift over the hand-engineered features. On all hold-out datasets, embeddings generated by the pretrained model show comparable performance to embeddings generated by finetuned models. This demonstrates the effectiveness of pretraining on a large corpus of diverse datasets and transferability of the pretrained model to significantly out-of-domain data.

5.4 Visualising the embedding space

To provide insights into the information encoded by embeddings, we provide visualisations of embeddings of merchant category code (MCC). These codes classify merchants and businesses by the type of goods or services provided. Large merchants classified as airlines, car rental companies and lodging providers typically have their own MCC. Since the input data does not provide any extra information relating these codes to each other, any emergent structure in the embedding space comes entirely from similarities in purchasing behaviors learnt by the model. Each MCC embedding was calculated as the average of all transaction embeddings corresponding to that

Table 5: Nearest neighbours in the embedding space of three MCC embeddings. For each nearest neighbour MCC we show cosine distance in the original space between the MCC in the top row.

	Lufthansa	Hilton Hotels	Fast Food
s	British Airways (0.23)	Doubletree Hotels (0.19)	Eating places (0.14)
neighbours	Scandinavian Airlines (0.32)	Hampton Inns	Convenience stores (0.26)
	Air France (0.33)	Fairmount Hotels (0.28)	Bakeries (0.27)
Nearest	Swissair (0.35)	Penta Hotels (0.29)	News Dealers (0.30)
	Turkish Airlines (0.38)	Marriott Hotels (0.32)	Drug Stores (0.35)

MCC. Figure 3 shows a t-SNE projection of the MCC embedding space together with three selected MCCs and their nearest neighbours. Table 5 lists the five nearest neighbours and their distances (measured by cosine distance) to each selected MCC.

We can see that the nearest neighbours of Lufthansa are all airline companies, while those of Hilton Hotels belong to the lodging industry, i.e. embeddings of merchants in the same industry are located close to each other in the embedding space. The cluster of embeddings for hotels is in close proximity to the cluster for airlines, which is expected since purchases from merchants in these two clusters are often correlated. This illustrates that generative modelling allows the embedding space to encode meaningful similarity between different MCCs akin to semantic similarity captured by word embeddings from large language models. This observation is consistent with [5], where merchant node embeddings from a graph neural network were used.

In a similar fashion, aggregations of transaction embeddings could be used to obtain embeddings of other entities, such as merchants, cardholders, transaction types, and geographical locations. They can potentially be used as features to support decision-making in recommendation engines and other financial systems.

6 CONCLUSIONS

In this paper, we present a self-supervised generative method for obtaining contextualised embeddings of financial transactions by combining two pretraining tasks: next event prediction and past reconstruction. Evaluations on publicly available datasets show that embeddings produced with this method outperform embeddings from other self-supervised methods and hand-engineered features on a range of downstream tasks. We apply our method to the card fraud detection problem and show that it significantly improves the value detection rate at high-precision thresholds. By pretraining on a large corpus of data from multiple issuing banks, we demonstrate that pretrained models trained with our method generalise well to significantly out-of-distribution data.

Pretraining generative models on large textual datasets has led to a class of Foundation Models that abstract away the complexity Skalski, et al.

of natural language modelling in modern AI applications. Likewise, our results on transaction sequences indicate that generative modelling encodes human purchasing behavior in a way that transfers effectively to diverse tasks and out-of-domain data. These properties may enable financial modelling applications to homogenize around a common component - a Foundation Model - which is trained on a large corpus of unlabelled data and which abstracts away the complexity of modelling financial behaviors. This motivates further research on questions of privacy, bias and the potential for few-shot learning, which we defer to future work.

ACKNOWLEDGMENTS

We thank Featurespace for the financial support and resources offered during the completion of this research. We also thank the NVIDIA Inception Program for generous support related to GPU hardware.

REFERENCES

- Sercan O. Arik and Tomas Pfister. 2020. TabNet: Attentive Interpretable Tabular Learning. arXiv:1908.07442 https://arxiv.org/abs/1908.07442
- [2] Dmitrii Babaev, Nikita Ovsov, İvan Kireev, Maria Ivanova, Gleb Gusev, Ivan Nazarov, and Alexander Tuzhilin. 2022. CoLES: Contrastive Learning for Event Sequences with Self-Supervision (SIGMOD '22). Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3514221.3526129
- [3] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. In Advances in Neural Information Processing Systems, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 12449–12460.
- [4] Tom Brown et al. 2020. Language Models are Few-Shot Learners. In Advances in Neural Information Processing Systems, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 1877–1901. https://proceedings.neurips.cc/paper_files/paper/2020/file/ 1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf
- [5] C. Bayan Bruss, Anish Khazane, Jonathan Rider, Richard Serpe, Antonia Gogoglou, and Keegan E. Hines. 2019. DeepTrax: Embedding Graphs of Financial Transactions. arXiv:1907.07225 [cs.LG] https://arxiv.org/abs/1907.07225
- [6] Mário Cardoso, Pedro Saleiro, and Pedro Bizarro. 2022. LaundroGraph: Self-Supervised Graph Representation Learning for Anti-Money Laundering. In Proceedings of the Third ACM International Conference on AI in Finance (New York, NY, USA) (ICAIF '22). Association for Computing Machinery, New York, NY, USA, 130–138. https://doi.org/10.1145/3533271.3561727
- [7] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. 2020. Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. In Proceedings of the 34th International Conference on Neural Information Processing Systems (Vancouver, BC, Canada) (NIPS'20). Curran Associates Inc., Red Hook, NY, USA, Article 831, 13 pages.
- [8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020.
 A Simple Framework for Contrastive Learning of Visual Representations. arXiv:2002.05709 https://arxiv.org/abs/2002.05709
- [9] Xinlei Chen and Kaiming He. 2021. Exploring Simple Siamese Representation Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 15750–15758.
- [10] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, Doha, Qatar, 1724–1734. https://doi.org/10.3115/v1/D14-1179
- [11] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In International Conference on Learning Representations. https://openreview.net/forum?id=rtxMH1BtvB
- [12] Alexis Conneau, Alexei Baevski, Ronan Collobert, Abdelrahman Mohamed, and Michael Auli. 2020. Unsupervised Cross-lingual Representation Learning for Speech Recognition. arXiv:2006.13979 https://arxiv.org/abs/2006.13979
- [13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 https://arxiv.org/abs/1810.04805
- [14] International Organization for Standardization. [n. d.]. ISO 8583-1:2003 Financial transaction card originated messages – Interchange message specifications –

- Part 1: Messages, data elements and code values. https://www.iso.org/standard/31628.html
- [15] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *Empirical Methods in Natural Language Processing (EMNLP)*. https://arxiv.org/abs/2104.08821
- [16] Antonia Gogoglou, Brian Nguyen, Alan Salimov, Jonathan Rider, and C. Bayan Bruss. 2020. Navigating the Dynamics of Financial Embeddings over Time. arXiv:2007.00591 https://arxiv.org/abs/2007.00591
- [17] Maitrey Gramopadhye, Shreyansh Singh, Kushagra Agarwal, Nitish Srivasatava, Alok Mani Singh, Siddhartha Asthana, and Ankur Arora. 2021. CuRL: Coupled Representation Learning of Cards and Merchants to Detect Transaction Frauds. In Artificial Neural Networks and Machine Learning ICANN 2021: 30th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 14–17, 2021, Proceedings, Part V (Bratislava, Slovakia). Springer-Verlag, Berlin, Heidelberg, 16–29. https://doi.org/10.1007/978-3-030-86383-8_2
- [18] Jean-Bastien Grill et al. 2020. Bootstrap Your Own Latent A New Approach to Self-Supervised Learning. In Advances in Neural Information Processing Systems, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 21271–21284. https://proceedings.neurips.cc/paper_files/paper/2020/file/f3ada80d5c4ee70142b17b8192b2958e-Paper.pdf
- [19] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In Advances in Neural Information Processing Systems, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/ paper files/paper/2017/file/5dd9db5e033da9c6fb5ba83c7a7ebea9-Paper.pdf
- [20] Xin Huang, Ashish Khetan, Milan Cvitkovic, and Zohar Karnin. 2020. TabTransformer: Tabular Data Modeling Using Contextual Embeddings. arXiv:2012.06678 https://arxiv.org/abs/2012.06678
- [21] Sanjeev Jha, Montserrat Guillen, and J. Christopher Westland. 2012. Employing transaction aggregation strategy to detect credit card fraud. Expert Systems with Applications 39, 16 (2012), 12650–12657. https://doi.org/10.1016/j.eswa.2012.05. 018
- [22] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *International Conference on Learning Representations*. https://openreview.net/forum?id=H1eA7AEtvS
- [23] Bertrand Lebichot, Fabian Braun, Olivier Caelen, and Marco Saerens. 2017. A graph-based, semi-supervised, credit card fraud detection system. In Complex Networks & Their Applications V, Hocine Cherifi, Sabrina Gaito, Walter Quattrociocchi, and Alessandra Sala (Eds.). Springer International Publishing, Cham, 721–733.
- [24] Naoto Minakawa, Kiyoshi Izumi, Hiroki Sakaji, and Hitomi Sano. 2022. Graph Representation Learning of Banking Transaction Network with Edge Weight-Enhanced Attention and Textual Information. In Companion Proceedings of the Web Conference 2022 (Virtual Event, Lyon, France) (WWW '22). Association for Computing Machinery, New York, NY, USA, 630–637. https://doi.org/10.1145/ 3487553.3524643
- [25] Ian Molloy, Suresh Chari, Ulrich Finkler, Mark Wiggerman, Coen Jonker, Ted Habeck, Youngja Park, Frank Jordens, and Ron van Schaik. 2017. Graph Analytics for Real-Time Scoring of Cross-Channel Transactional Fraud. In Financial Cryptography and Data Security, Jens Grossklags and Bart Preneel (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 22–40.
- [26] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei Efros. [n. d.]. Context Encoders: Feature Learning by Inpainting. In Computer Vision and Pattern Recognition (CVPR). https://doi.org/10.1109/CVPR.2016.278
- [27] Alec Radford and Ilya Sutskever. 2018. Improving Language Understanding by Generative Pre-Training. In arxiv. https://cdn.openai.com/research-covers/ language-unsupervised/language_understanding_paper.pdf
- [28] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. (2019). https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf
- [29] Gowthami Somepalli, Avi Schwarzschild, Micah Goldblum, C. Bayan Bruss, and Tom Goldstein. 2022. SAINT: Improved Neural Networks for Tabular Data via Row Attention and Contrastive Pre-Training. https://openreview.net/forum?id= nl.2IDIsrZU
- [30] Trieu H. Trinh, Minh-Thang Luong, and Quoc V. Le. 2019. Selfie: Self-supervised Pretraining for Image Embedding. arXiv:1906.02940 https://arxiv.org/abs/1906. 02940
- [31] Talip Ucar, Ehsan Hajiramezanali, and Lindsay Edwards. 2021. SubTab: Subsetting Features of Tabular Data for Self-Supervised Representation Learning. In Advances in Neural Information Processing Systems, M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (Eds.), Vol. 34. Curran Associates, Inc., 18853–18865.
- [32] Rafaël Van Belle, Sandra Mitrović, and Jochen De Weerdt. 2020. Representation Learning in Graphs for Credit Card Fraud Detection. In Mining Data for Financial Applications, Valerio Bitetta, Ilaria Bordino, Andrea Ferretti, Francesco Gullo, Stefano Pascolutti, and Giovanni Ponti (Eds.). Springer International Publishing,

- Cham, 32-46.
- [33] Rafaël Van Belle, Charles Van Damme, Hendrik Tytgat, and Jochen De Weerdt. 2022. Inductive Graph Representation Learning for fraud detection. Expert Systems with Applications 193 (2022), 116463. https://doi.org/10.1016/j.eswa.2021. 116463
- [34] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2019. Representation Learning with Contrastive Predictive Coding. arXiv:1807.03748 https://arxiv.org/abs/1807. 03748
- [35] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. 2019. CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features. In *International Conference on Computer Vision (ICCV)*.
- [36] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stephane Deny. 2021. Barlow Twins: Self-Supervised Learning via Redundancy Reduction. In Proceedings of the 38th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 139), Marina Meila and Tong Zhang (Eds.). PMLR, 12310– 12320. https://proceedings.mlr.press/v139/zbontar21a.html
- [37] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond Empirical Risk Minimization. In International Conference on Learning Representations. https://openreview.net/forum?id=r1Ddp1-Rb
- [38] Jiachen Zhu, Rafael M. Moraes, Serkan Karakulak, Vlad Sobol, Alfredo Canziani, and Yann LeCun. 2022. TiCo: Transformation Invariance and Covariance Contrast for Self-Supervised Visual Representation Learning. arXiv:2206.10698 [cs.CV]