Supplementary Information for Predicting Company Growth using Scaling Theory Informed Machine Learning

Ruyi Tao[1,2], Veronica R. Cappelli[3], Kaiwei Liu[1,2], Marcus J. Hamilton[4,5], Christopher P. Kempes[4], Geoffrey B. West[4], Jiang Zhang[1,2*]

[1]Department of Systems Science, Beijing Normal University, Beijing, China.
[2]Swarma Research, Beijing, China.
[3]Department of Managerial Decision Sciences, IESE Business School, Barcelona, Spain.
[4]The Santa Fe Institute, Santa Fe, USA.
[5]Department of Anthropology and School of Data Science, University of Texas at San Antonio, San Antonio, USA.

*Corresponding author(s). E-mail(s): zhangjiang@bnu.edu.cn;
Contributing authors: taoruyi@mail.bnu.edu.cn;

S1 Data Preprocessing

We screened and preprocessed the raw data as follows:

1. **Feature Selection**: First, we removed the features whose missing values were greater than 50%.
2. **Noise reduction**: We filtered out companies with a time series of less than 2 years. Companies that appear in the dataset for only 1 or 2 years can neither serve as training samples nor be predictable. And then, we also filtered out the years in which abnormal data appear in financial statements, such as when liabilities, income costs, etc., are negative or zero. This type of data is considered anomalous and needs to be eliminated.
3. **Missing value imputation**: There are some years with missing data in the dataset, and the previous step of anomaly handling may also leave spots where data

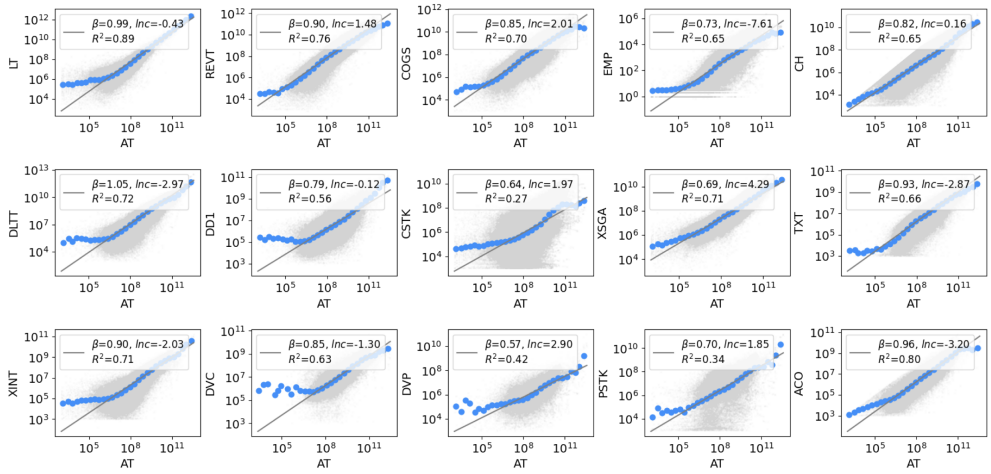

Fig. S1 We have 23 financial indicators in total (See Table S1 in SI). Among them, 15 indicators are positive and exhibit scaling relationships with assets. Therefore, at most 16 predictors can be used, including assets itself.

are missing. In this paper's experiments, if the missing value is in the middle, we used the average of the time series data before and after to replace it; if the missing value is at the endpoints, we used the previous or subsequent value to replace it.

4. **Inflation adjustment**: We applied inflation adjustments to all the monetary data across different years, with 2019 serving as the base year. The adjustments were made via the year-average inflation consumer price (ICP).
5. **Logarithm transformation**: We took the natural logarithm of all monetary data for scaling. For some indicators that may have negative values (such as profit, cash flow, etc., data from the cash flow statement) and some values that occasionally occur as 1 (such as the number of employees), we have developed a linear-log method in practice, as shown in Equation S1.

$$f(x) = \delta(x)\ln(|x|+1), x \neq 0 \tag{S1}$$

$\delta(x)$ represents the sign of x, indicating whether it is positive or negative. This method can achieve the same effect as taking the logarithm: not only can it scale negative values similarly, but it can also be used within the 0-1 interval.

S1.1 Sample split: Training, validation, and test data

We first split the data between post-2010 and pre-2010 and then divided the pre-2010 observations into training, validation, and testing sets at a 6:2:2 ratio according to the number of companies. That is, over 13,000 companies constitute in the training set, over 5,000 companies constitute the validation set, and over 5,000 companies constitute the test set. All the post-2010 data are also used for testing. Figure S3 shows a schematic diagram of the dataset division. The reason for this approach is that,

Table S1 Financial Indicators

Indicators	Description	Minimum	Maximum
AT	Assets	1.0240×10^{6}	3.7712×10^{12}
LT	Liabilities	1.0460×10^{3}	3.7741×10^{12}
REVT	Revenue	7.6300×10^{2}	5.1479×10^{11}
COGS	Cost of Goods Sold	0	3.9934×10^{11}
EMP	Employee	1	2.3000×10^{6}
CH	Cash	0	1.6903×10^{11}
DLTT	Long-Term Debt	0	3.5579×10^{12}
DD1	Long-Term Debt Due in One Year	0	1.9923×10^{11}
CSTK	Common/Ordinary Stock (Capital)	0	7.1649×10^{10}
XSGA	Selling, General and Administrative Expenses	0	1.0729×10^{11}
TXT	Income Taxes	0	4.8919×10^{10}
XINT	Interest and Related Expenses	0	1.6136×10^{11}
DVC	Dividends Common/Ordinary	0	4.9146×10^{10}
DVP	Dividends - Preferred/Preference	0	9.3423×10^{10}
PSTK	Preferred/Preference Stock (Capital)	0	1.5213×10^{11}
ACO	Current Assets - Other	0	6.5765×10^{10}
AQI	Acquisitions - Income Contribution	-1.4806×10^{8}	6.5765×10^{10}
AQS	Acquisitions - Sales Contribution	-6.1114×10^{9}	9.4562×10^{11}
MII	Minority Interest (Income Account)	-1.4352×10^{10}	1.3179×10^{10}
CEQ	Common/Ordinary Equity	-1.5700×10^{11}	3.6440×10^{11}
EBIDTA	Earnings Before Interest	-9.2932×10^{10}	1.4612×10^{11}
NI	Net Income (Loss)	-1.3997×10^{11}	1.2225×10^{11}
RE	Return Earnings	-1.4926×10^{11}	4.1977×10^{11}

unlike typical time series forecasting tasks, company-level financial statement data is characterized by having a large number of samples but relatively short time spans. Therefore, we choose to partition the dataset primarily based on the characteristics of the data. In most cases, different batches correspond to different companies rather than different time windows.

The fitting of parameters for $GM(\beta_X, c_X)$ and the training of the machine learning are performed on the training set, which is shown the blue section of Figure S3.

S1.2 Choice of Hyperparameters

Length of historical time-series is set as $S = 3$ and also the prediction length $T = 3$ for the most cases. The mean squared error (MSE) is chosen as the loss function, and the Adam optimizer is applied for gradient descent, with an adaptive learning rate and a weight decay of 0.005. Unless otherwise specified, the hidden size of the MLP in both the encoder and decoder is set to 32, and to 8 for iTransformer. All attention layers are configured with a single layer.

S2 Detail of iTransformer Model

The input of encoder module is $\mathbf{X}_{t-S:t,:}$, including the target variables $\mathbf{X}_{t-S:t,:N}$ and concatenates them with other potentially useful predictive variables, such as macroeconomic information $\mathbf{X}^{macro}_{t-S:t,:} \in \mathbb{R}^{S\times(N'-N)}$. Ultimately, this part is modeled as follows:

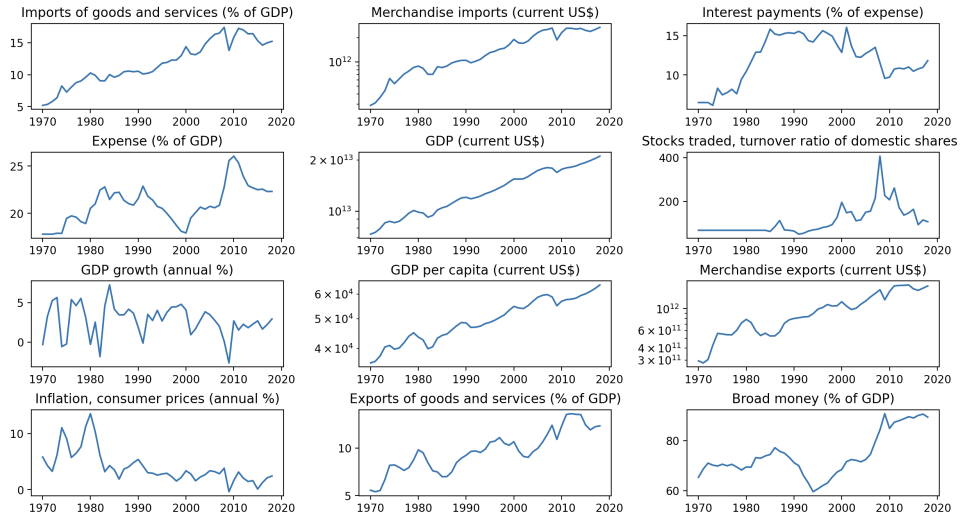

Fig. S2 12 Macroeconomic Indicators from 1970 to 2019.

$$\begin{aligned} H_0 &= Embedding(\mathbf{X}_{t-S:t,:}), \\ H_{l+1} &= Encoder(H_l), l = 0, \ldots, L-1 \\ En_{l+1} &= Projection(H_L) \end{aligned} \tag{S2}$$

Embedding and *Projection* are both linear transformation here. $\mathbf{X}$ is normalized before embedding. H_l is the hidden state for the lth layer. For the MLP model, the encoder module consists of linear layers followed by activation functions. For the iTransformer architecture, we largely adhere to the settings described in [1], including multivariate attention , 2 layer normalization, and feed-forward networks. Detailed architectures of both MLP and iTransformer are illustrated in Figure S4. The final output denote as En_{l+1} is passed into the decoder module.

In the decoder module, we predict future time steps from t to $t+T$. The inputs include $\mathbf{X}_{t:t+T}{}^{GM}$, which is the prediction from the GM, macroeconomic information $\mathbf{X}_{t:t+T}{}^{macro}$ and, of course, the output from the encoder modules En_{l+1}.

$$\mathbf{O} = Decoder(\mathbf{X}^{GM}{}_{t:t+T} \bigoplus \mathbf{X}^{macro}{}_{t:t+T}, En_{l+1}) \tag{S3}$$

$\bigoplus$ is the concatenating operation. For the MLP model, the decoder module similarly comprises linear layers with activation functions. Regarding the iTransformer architecture, since the original model lacks this component, we introduce an additional attention module to compute the attention matrix between extra input and En_{l+1}. Detailed architectures are illustrated in Figure S4. After reshaping, the output of decoder module is $\mathbf{O} = \{\mathbf{o}_1, \ldots, \mathbf{o}_T\} \in \mathbb{R}^{T \times N}$ after adding back the output of GM $\mathbf{X}^{GM}_{t:t+T}$, we can obtain the final prediction $\hat{\mathbf{Y}}_{t:t+T} = \mathbf{O}_{t:t+T} + \mathbf{X}^{GM}_{t:t+T}$.

The loss function here is the mean square error (MSE), and it optimizes the difference between the prediction $\hat{\mathbf{Y}}$ and the actual data $\mathbf{Y}$ in log space.

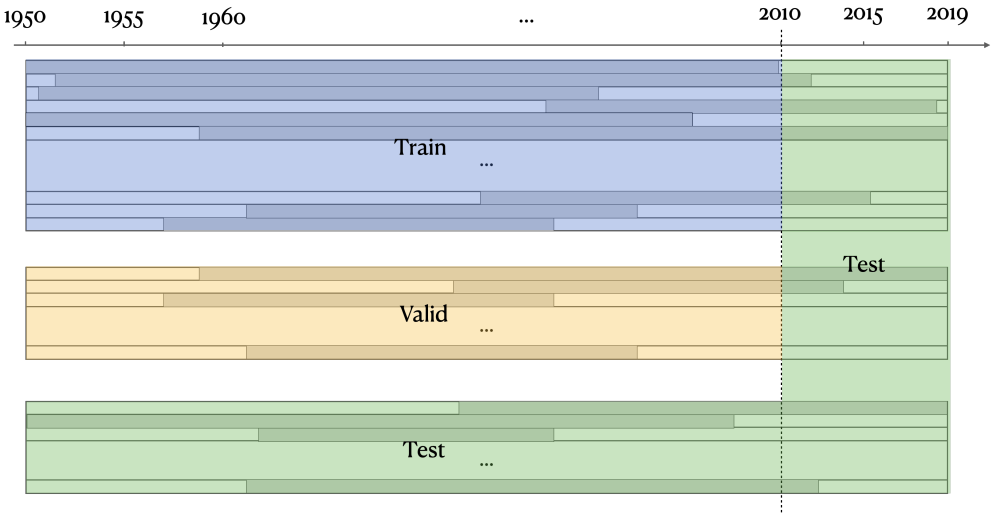

Fig. S3 Data setting. Our dataset spans from 1950 to 2019. In the figure, each gray row represents a company, and the solid-colored segment indicates the years during which the company was active. We first partitioned the data into periods before and after 2010 (dash line). Companies established before 2010 were further divided into a training set (blue), validation set (yellow), and test set (green). In addition, all observations from 2010 onward were included in the test set.

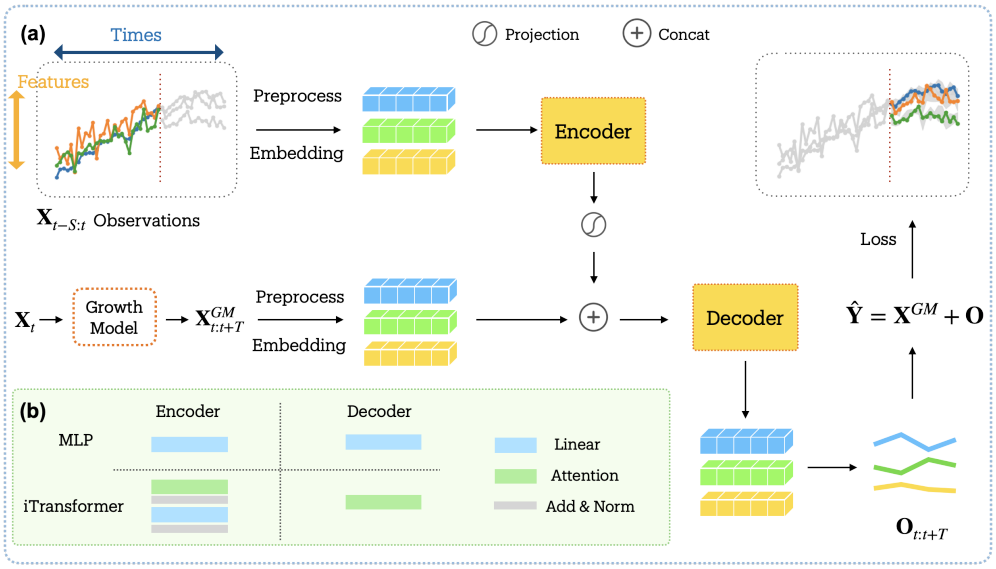

Fig. S4 Overall architecture of GM combined neural network. (a) presents the complete framework from raw data processing to final prediction results; (b) illustrates the core components of both MLP and iTransformer architectures.

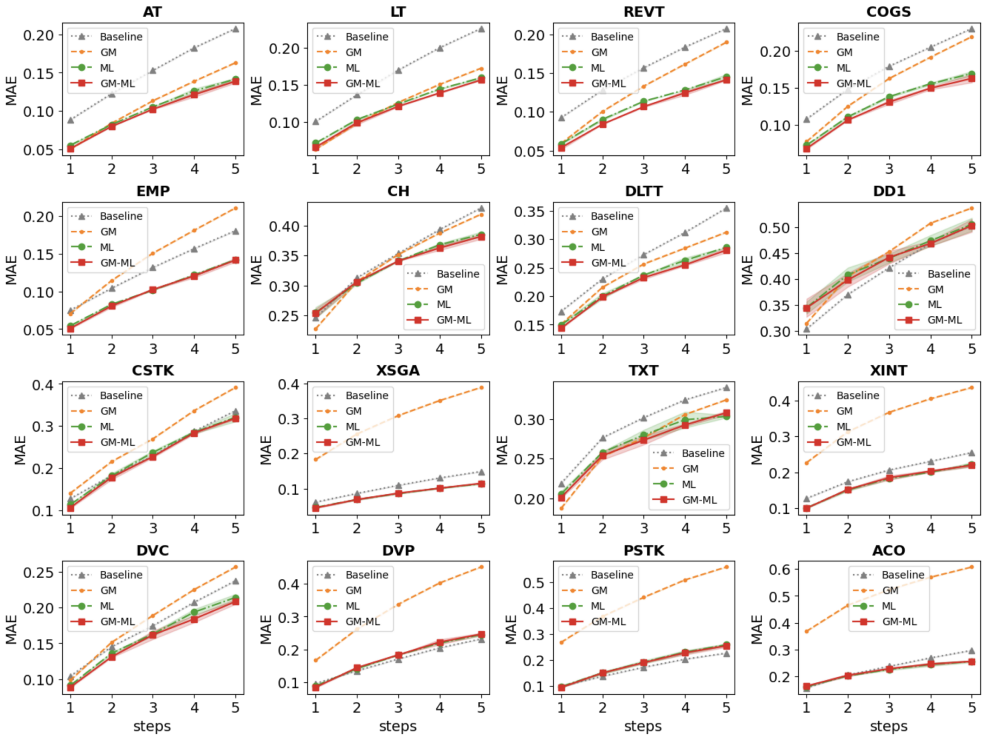

Fig. S5 5-step-ahead MAE comparison across models for all predicted variables

$$Loss = \frac{1}{T}\Sigma_j^T || \ln Y_j - \ln \hat{Y}_j ||^2 \tag{S4}$$

References

[1] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024.

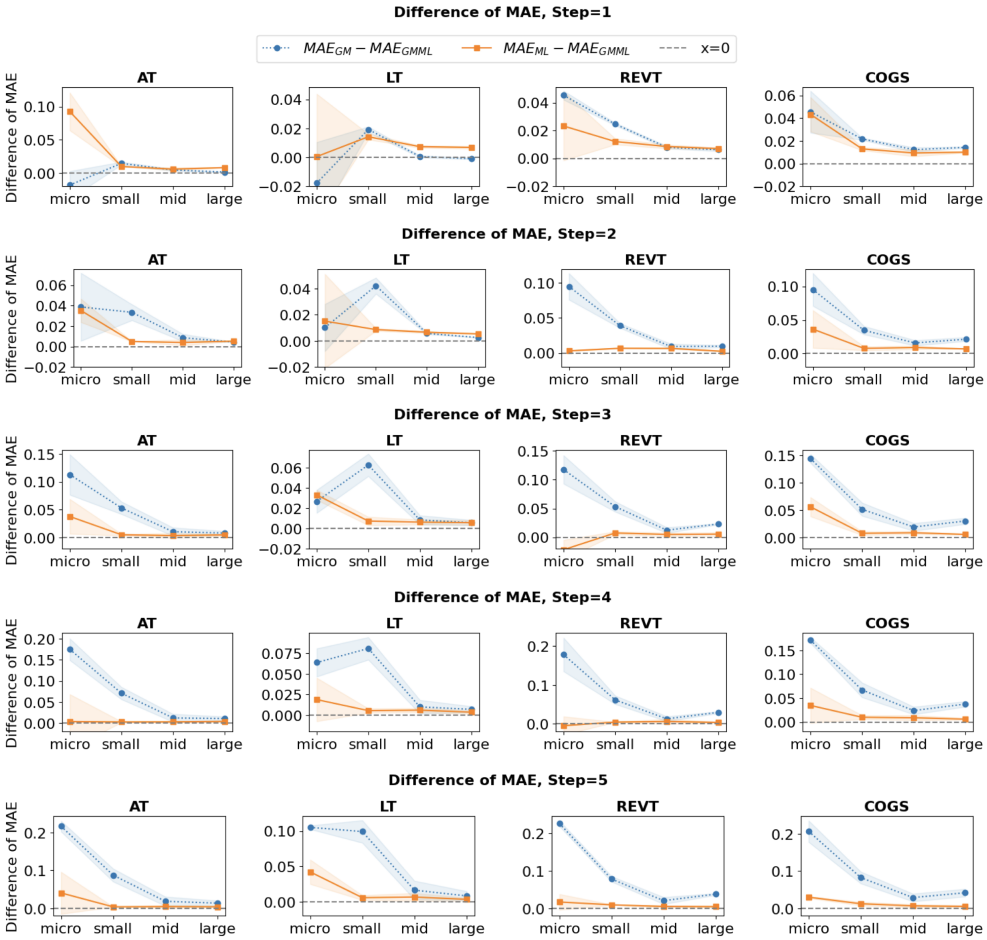

Fig. S6 Visualization of performance gain across size groups for 5-step-ahead predictions

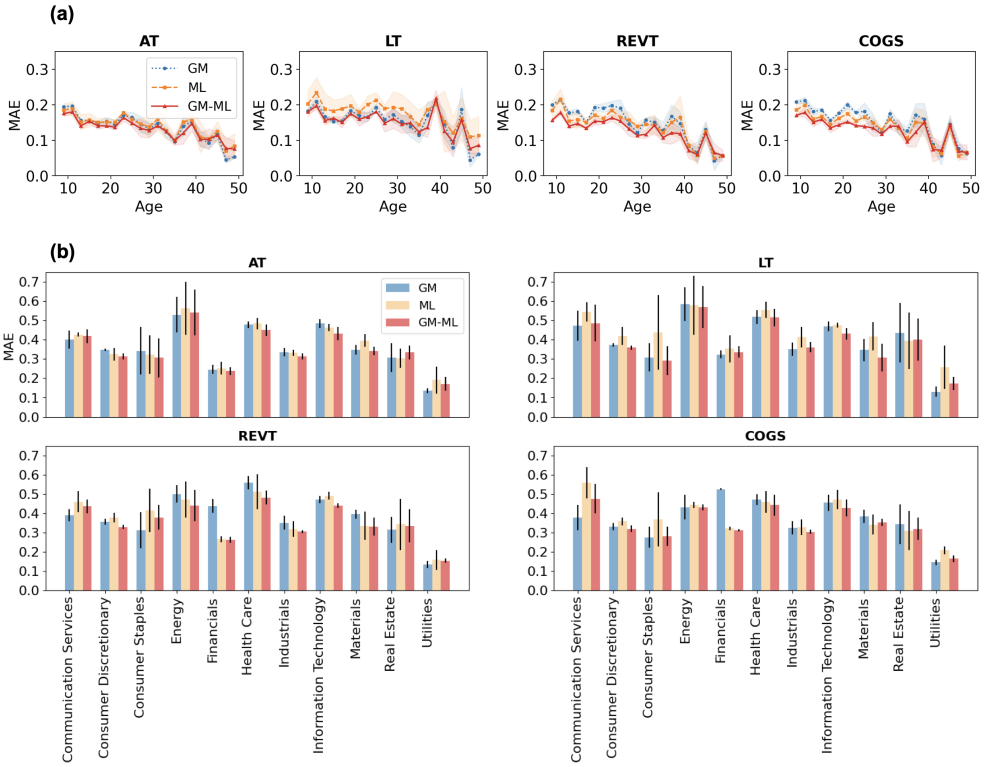

Fig. S7 (a).Comparison of MAE across different age groups of companies for different models. GM is solid blue, ML is dashed yellow and GM-ML is solid red line. The columns represent different prediction time steps. (b). Comparison of MAE across different sectors for GM-ML.