# JP Morgan MLCOE TSRL 2026 Internship Question 1: Application for lending department of a bank


by

**REN Haijie**


November 2025, Hong Kong


THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

# TABLE OF CONTENTS

# CHAPTER 1

# Problem: The Challenge of Balance Sheet Forecasting

## 1.1 Motivation and Context

Forecasting financial statements is a cornerstone of corporate finance, valuation, and credit risk assessment. Traditional budgeting textbooks often construct pro forma financial statements by scaling historical figures with forecasted sales and then "closing" the balance sheet using a *plug* variable. A plug is a residual item—typically Cash, Debt, or Equity—used to force the accounting equation to hold. While convenient, this practice masks modeling errors: even if the underlying assumptions are flawed, the balance sheet will appear to balance because the plug absorbs the inconsistencies [1].

In contrast, the "No-Plug, No-Circularity" approach advocated by Vélez-Pareja et al. constructs all balance sheet items as explicit results of underlying transactions. The goal is to create forecasts that are internally consistent by design, where any mismatch reveals a genuine modeling mistake rather than being hidden by a residual calculation [1, 2].

## 1.2 Financial Statements: Stocks and Flows

From a modeling perspective, the firm is described by three interrelated statements: the Balance Sheet (BS), the Income Statement (IS), and the Cash Flow Statement (CB).

- **Stock Variables (BS):** Variables measured at a specific point in time $t$ (e.g., Cash, Inventory, Debt, Equity).
- **Flow Variables (IS, CB):** Variables measured over a period $[t, t+1]$ (e.g., Sales, Expenses, Interest, CapEx, Repayments).

Let $t = 0, 1, \dots, T$ index discrete time periods. The fundamental *accounting identity* must hold for all $t$ [1]:

$$\text{Assets}_t = \text{Liabilities}_t + \text{Equity}_t \qquad \forall t = 0, \dots, T, \tag{1.1}$$

Crucially, the link between periods is governed by the *Clean Surplus Relation* (or flow-to-stock accumulation). For Retained Earnings (RE), this is defined as:

$$\mathrm{RE}_{t+1} = \mathrm{RE}_t + \mathrm{NI}_{t+1} - \mathrm{Div}_{t+1} \tag{1.2}$$

where NI is Net Incomeand Div are dividends. This implies that changes in equity must be fully explained by income generation and payout policies, not by arbitrary adjustments. [2]

## 1.3 Plug-Based Models and the Circularity Problem

In many spreadsheet-based planning models, the balancing problem of the forecasted balance sheet is addressed by defining a plug variable. Typical choices include:

$$\mathrm{Cash}_t = \text{Total Liabilities}_t + \mathrm{Equity}_t - \text{Non-cash Current Assets}_t - \text{Net Fixed Assets}_t, \tag{1.3}$$

$$\mathrm{Debt}_t = \text{Total Assets}_t - \text{Current Liabilities}_t - \mathrm{Equity}_t, \tag{1.4}$$

or similar formulations where one item is defined purely as a residual. In these constructions, changes in the plug are not explicitly linked to actual cash inflows and outflows, debt issuance and repayment, or equity transactions; instead, they are whatever is needed to satisfy (1.1). As a result, the balance sheet may mask conceptual or numerical errors in the modelling of all other items.[1, 2]

A related but distinct issue is the *circularity problem* in discounted cash flow (DCF) valuation. In levered valuation frameworks, the discount rate (e.g. WACC or cost of equity) depends on leverage and therefore on the firm's value, while the firm's value itself is computed by discounting cash flows using that discount rate. This simultaneity leads to circularity between value, leverage and the cost of capital [3]. Traditional solutions rely on iterative "rolling WACC" procedures or target leverage assumptions, but these can be conceptually unsatisfactory and numerically fragile [3].

For the purposes of this project, these two problems—plug-based balancing of financial statements and circularity between value and cost of capital—highlight the need for models that:

- respect accounting identities such as (1.1) and (1.2) by construction;
- derive all balance sheet items from explicit transaction flows instead of residual plugs;
- avoid circular dependencies by specifying a consistent timing convention for interest, taxes, and financing decisions.

## 1.4 Dynamic Time-Series Formulation

Following the dynamic microsimulation framework of Shahnazarian (2004) [4], we view the balance sheet evolution as a system of difference equations driven by firm-level decisions and macroeconomic conditions. This leads to a general dynamic system specification:

$$\mathbf{y}_{t+1} = f\left(\mathbf{y}_t, \mathbf{x}_t, \mathbf{z}_t\right) + \boldsymbol{\varepsilon}_{t+1} \tag{1.5}$$

Where:

- $\mathbf{y}_t$: Vector of Balance Sheet stocks (State variables).
- $\mathbf{x}_t$: Vector of flow variables (Income Statement and Cash Flow items).
- $\mathbf{z}_t$: Exogenous covariates (e.g., Interest rates, GDP growth).
- $\boldsymbol{\varepsilon}_{t+1}$ is a disturbance term.

## 1.5 Formal Statement of the Problem

Based on the prompt and the theoretical context, the modeling problem can be formally stated as:

> **Input:** A historical time series of financial statements $\left\{(\mathrm{BS}_t, \mathrm{IS}_t, \mathrm{CF}_t)\right\}_{t=0}^{T_{\text{hist}}}$, augmented with macroeconomic indicators $\mathbf{z}_t$.
>
> **Output:** A multi-period forecast of future balance sheets $\left\{\widehat{\mathrm{BS}}_t\right\}_{t=T_{\text{hist}}+1}^{T_{\text{horizon}}}$ such that:
>
> 1. **Consistency:** The forecast respects the accounting identity (Eq. 1.1) and clean surplus relation (Eq. 1.2) at every time step.
> 2. **Causality:** All balance sheet items are derived from explicit transaction flows (operating, investing, financing) rather than residual plugs.
> 3. **Dynamics:** The evolution follows a dynamic model (Eq. 1.5) capable of capturing non-linear dependencies, implementable in TensorFlow/Python.

# CHAPTER 2

# A Simple Dynamic Balance Sheet Model

## 2.1 Aggregated Balance Sheet Structure

To keep the notation tractable, we work with an aggregated balance sheet. At the end of period $t$, we collect the main stock variables into the following categories:

- Cash and cash equivalents: $C_t$;
- Accounts receivable: $\mathrm{AR}_t$;
- Inventories: $\mathrm{Inv}_t$;
- Net property, plant and equipment: $K_t$;
- Accounts payable: $\mathrm{AP}_t$;
- Interest-bearing debt (short- and long-term combined): $D_t$;
- Equity (book value): $E_t$.

Total assets and total liabilities plus equity are then

$$\mathrm{Assets}_t = C_t + \mathrm{AR}_t + \mathrm{Inv}_t + K_t, \tag{2.1}$$

$$\mathrm{Liab}_t + E_t = \mathrm{AP}_t + D_t + E_t. \tag{2.2}$$

The double-entry principle requires the accounting identity

$$C_t + \mathrm{AR}_t + \mathrm{Inv}_t + K_t = \mathrm{AP}_t + D_t + E_t, \qquad \forall t. \tag{2.3}$$

Our goal is to specify simple dynamic equations for the evolution of these balance-sheet items that:

1. are consistent with standard accounting identities and the clean surplus relation;
2. avoid the use of residual "plugs";
3. can be written in a compact time-series form $\mathbf{y}_{t+1} = f(\mathbf{y}_t, \mathbf{x}_t)$.

## 2.2 Income Statement and Flow Variables

We consider a stylised income statement for period $t$:

$$\text{Sales} : S_t, \tag{2.4}$$

$$\text{Cost of goods sold} : \text{COGS}_t, \tag{2.5}$$

$$\text{Operating expenses} : \text{OPEX}_t, \tag{2.6}$$

$$\text{Depreciation} : \text{Dep}_t, \tag{2.7}$$

$$\text{Interest expense} : \text{Int}_t, \tag{2.8}$$

$$\text{Tax expense} : \text{Tax}_t, \tag{2.9}$$

$$\text{Net income} : \text{NI}_t. \tag{2.10}$$

We define the usual relationships

$$\text{GrossIncome}_t = S_t - \text{COGS}_t, \tag{2.11}$$

$$\text{EBIT}_t = \text{GrossIncome}_t - \text{OPEX}_t - \text{Dep}_t, \tag{2.12}$$

$$\text{Int}_t = r_D D_{t-1}, \tag{2.13}$$

$$\text{EBT}_t = \text{EBIT}_t - \text{Int}_t, \tag{2.14}$$

$$\text{Tax}_t = \tau \max\{\text{EBT}_t, 0\}, \tag{2.15}$$

$$\text{NI}_t = \text{EBT}_t - \text{Tax}_t, \tag{2.16}$$

where $r_D$ is the effective interest rate on debt and $\tau$ is the corporate tax rate. Using the clean surplus relation, book equity evolves according to

$$E_t = E_{t-1} + \text{NI}_t - \text{Div}_t + \text{EquityIssues}_t, \tag{2.17}$$

with $\text{Div}_t$ dividends and $\text{EquityIssues}_t$ net new equity raised in period $t$. A simple policy is to assume a constant payout ratio $p_{\text{div}}$:

$$\text{Div}_t = p_{\text{div}} \text{NI}_t, \tag{2.18}$$

and treat $\text{EquityIssues}_t$ as determined by a financing rule (e.g. a target leverage ratio). In the implementation, we assume that the company does not pay dividends when the net profit is negative, i.e., $Div_t = \max(p_{div} NI_t, 0)$.

## 2.3 Working-Capital Policies

Following standard financial forecasting practice and the no-plug approach [1, 2], we derive working-capital accounts from operational flows instead of using residual balancing items.

We assume simple linear policies:

$$\mathrm{AR}_t = \phi_{\mathrm{AR}}\, S_t, \tag{2.19}$$

$$\mathrm{Inv}_t = \phi_{\mathrm{Inv}}\, \mathrm{COGS}_t, \tag{2.20}$$

$$\mathrm{AP}_t = \phi_{\mathrm{AP}}\, \mathrm{COGS}_t, \tag{2.21}$$

where $\phi_{\mathrm{AR}}$, $\phi_{\mathrm{Inv}}$ and $\phi_{\mathrm{AP}}$ are policy parameters analogous to days-sales-outstanding and days-payables-outstanding. In differential form, the changes in these accounts between periods $t-1$ and $t$ are

$$\Delta\mathrm{AR}_t = \mathrm{AR}_t - \mathrm{AR}_{t-1}, \tag{2.22}$$

$$\Delta\mathrm{Inv}_t = \mathrm{Inv}_t - \mathrm{Inv}_{t-1}, \tag{2.23}$$

$$\Delta\mathrm{AP}_t = \mathrm{AP}_t - \mathrm{AP}_{t-1}. \tag{2.24}$$

## 2.4 Fixed Assets and Debt Dynamics

Let $I_t$ denote gross capital expenditures (CAPEX) in period $t$ and $\delta$ the depreciation rate on fixed assets. Net PPE evolves as

$$K_t = (1-\delta)K_{t-1} + I_t. \tag{2.25}$$

For interest-bearing debt, a simple one-factor debt stock $D_t$ suffices for illustration. Let $\mathrm{NewDebt}_t$ be new borrowings and $\mathrm{Repay}_t$ principal repayments in period $t$. Then

$$D_t = D_{t-1} + \mathrm{NewDebt}_t - \mathrm{Repay}_t. \tag{2.26}$$

The corresponding interest expense in period $t$ is computed from the opening balance $D_{t-1}$ as in the income statement.

## 2.5 Cash Flow Identity and Cash Dynamics

The cash account closes the system. Using the standard decomposition of cash flows, net cash flow in period $t$ can be written as:

$$\mathrm{NCF}_t = \underbrace{\mathrm{NI}_t + \mathrm{Dep}_t - \Delta \mathrm{AR}_t - \Delta \mathrm{Inv}_t + \Delta \mathrm{AP}_t}_{\text{Operating cash flow}}$$

$$- \underbrace{I_t}_{\text{Investing (CAPEX)}} + \underbrace{\mathrm{NewDebt}_t - \mathrm{Repay}_t + \mathrm{EquityIssues}_t - \mathrm{Div}_t}_{\text{Financing cash flow}}. \quad (2.27)$$

The cash balance then evolves according to

$$C_t = C_{t-1} + \mathrm{NCF}_t. \quad (2.28)$$

Equations (2.17)–(2.28), together with the working capital policies (2.21) and the fixed-asset and debt dynamics (2.25)–(2.26), define the evolution of all stock variables on the balance sheet. Because each change is tied to explicit flows and the cash-flow identity (2.27) holds, the balance sheet identity (2.3) is satisfied automatically at each $t$ without the use of plugs.

## 2.6 Time-Series Representation and Accounting Constraints

Collect all balance-sheet stock variables into a state vector

$$\mathbf{y}_t = \begin{bmatrix} C_t & \mathrm{AR}_t & \mathrm{Inv}_t & K_t & \mathrm{AP}_t & D_t & E_t \end{bmatrix}^\top, \quad (2.29)$$

and let

$$\mathbf{x}_t = \begin{bmatrix} S_t & \mathrm{COGS}_t & \mathrm{OPEX}_t & I_t & \mathrm{EquityIssues}_t & \mathrm{NewDebt}_t & \mathrm{Repay}_t \end{bmatrix}^\top \quad (2.30)$$

collect the main flow variables and financing decisions during period $t$. For fixed policy parameters

$$\theta = (\phi_{\mathrm{AR}}, \phi_{\mathrm{Inv}}, \phi_{\mathrm{AP}}, \delta, r_D, \tau, p_{\mathrm{div}}),$$

the deterministic version of the model can be written compactly as

$$\mathbf{y}_t = f(\mathbf{y}_{t-1}, \mathbf{x}_t; \theta), \quad (2.31)$$

where $f$ is implicitly defined by the accounting equations above. A stochastic version introduces an additive disturbance term:

$$\mathbf{y}_t = f(\mathbf{y}_{t-1}, \mathbf{x}_t; \theta) + \mathbf{n}_t, \quad (2.32)$$

which fits the generic form $y(t+1) = f(x(t), y(t)) + n(t)$ suggested in the assignment.

Crucially, not all components of $\mathbf{y}_t$ are independent: the accounting identity (2.3) imposes a linear constraint of the form

$$h(\mathbf{y}_t) := C_t + \mathrm{AR}_t + \mathrm{Inv}_t + K_t - \mathrm{AP}_t - D_t - E_t = 0, \quad \forall t. \tag{2.33}$$

Any statistical or machine-learning model that forecasts $\mathbf{y}_t$ must respect this constraint to produce economically meaningful balance sheets. This simple dynamic model can be a structural prior for the machine learning methods.

In the next chapter, we will perform a literature review and try to answer the questions "Is it possible to model this problem as a time series?" and "How do we handle the accounting identities?"

# CHAPTER 3

# Literature Review: Methods for Forecasting Balance Sheets

## 3.1 Overview

The problem of forecasting financial statements, specifically the balance sheet, has been approached from two complementary angles. On the one hand, traditional corporate finance and accounting literature develops *structural*, transaction-based models that guarantee internal consistency of the financial statements. On the other hand, econometric and machine-learning research provides *data-driven* time-series models that can capture complex temporal dynamics but typically ignore accounting identities.

Building on the problem formulation in Chapter 1 and the simple dynamic balance sheet model in Chapter 2, this chapter reviews the main strands of literature relevant for Part 1 of the assignment:

1. accounting-driven forecasting models such as percentage-of-sales and turnover-ratio methods;
2. plug-free, constraint-aware deterministic frameworks following Vélez-Pareja and co-authors;
3. classical time-series econometric approaches;
4. modern machine-learning sequence models (RNNs, Transformers, TCNs);
5. recent work on constraint-aware neural networks that combine the two worlds.

The goal is to motivate my choice of a *constraint-aware LSTM* model that treats the balance sheet as a multivariate time series while enforcing accounting identities at each forecast step.

## 3.2 Accounting-Driven Models (Percentage-of-Sales, Turnover Ratios)

Accounting-driven approaches rely on explicit financial relationships and managerial assumptions rather than learning from data patterns. The most ubiquitous technique is the *percentage-of-sales* *method*, where future income-statement and balance-sheet items are forecast by scaling historical ratios to projected sales [5, 6]. For example, accounts receivable, inventories, and accounts payable may be modelled as fixed percentages of sales or cost of goods sold (COGS).

Closely related are models based on *turnover ratios*, such as days sales outstanding (DSO), days inventory outstanding (DIO), and days payables outstanding (DPO), which explicitly link working-capital accounts to operational flows [6, 7]. These methods can be integrated into spreadsheet-based pro forma models where the analyst specifies drivers (e.g. sales growth, margins, capital expenditure) and uses ratio-based rules to derive the balance sheet.

These techniques are intuitive, transparent, and widely taught in corporate finance and accounting textbooks [4]. However, they typically address the balance-sheet balancing problem by introducing a *plug variable*, such as cash, debt, or equity. After forecasting all other items, the plug is computed residually to satisfy the accounting identity

$$\text{Assets}_t = \text{Liabilities}_t + \text{Equity}_t.$$

While convenient, this practice can hide modelling errors: even if some drivers or ratios are poorly specified, the balance sheet will still "balance" because the plug absorbs inconsistencies. Moreover, plugs can generate circularity—for example when interest expense depends on debt, which in turn is determined as a residual from the balance sheet [1].

## 3.3 Constraint-Aware Deterministic Frameworks

To address the limitations of plug-based models, several authors develop fully articulated, plug-free forecasting frameworks in which all variables are driven by explicit transactions and policies rather than residual balancing items. Vélez-Pareja and co-authors propose a *no-plug, no-circularity* approach in which:

- all balance-sheet accounts are defined as accumulations of past flows;
- the cash budget is derived from operating, investing, and financing cash flows;
- debt schedules, interest expenses, and tax shields are computed sequentially, respecting the timing of flows [1, 8].

In this framework, the equity account is driven by the clean-surplus relation

$$E_t = E_{t-1} + \text{Net Income}_t - \text{Dividends}_t + \text{Equity Issues}_t,$$

and debt dynamics are specified through explicit issuance and repayment policies. Circularity between firm value, leverage, and the cost of capital in discounted cash flow (DCF) valuation is replaced by consistent timing conventions and, where necessary, explicit iterative algorithms that converge to a fixed point [5]. As a result, any violation of the accounting identity is indicative of a genuine modelling error rather than being mechanically rectified by a plug.

Shahnazarian's work on dynamic microsimulation of firms provides a broader methodological perspective [4]. Firms are modelled as evolving agents whose stock variables (balance sheet) follow difference equations driven by flow variables (income statement and cash flows), economic conditions, and policy variables. This naturally leads to a system of dynamic equations for balance-sheet items, forming the conceptual bridge toward viewing the problem as a time-series forecasting task.

These deterministic frameworks are highly suited to scenario analysis and stress testing: given assumptions on growth, margins, and policy variables, they produce internally consistent pro forma financial statements. However, they do not automatically adapt to data in the way statistical or machine-learning models do; instead, they rely on the analyst's judgment to specify trajectories for key drivers.

## 3.4 Time Series Econometric Methods

A complementary line of research treats financial statement items as stochastic time series and applies classical econometric tools. Popular approaches include:

- univariate ARIMA models for individual accounts (e.g. sales, total assets, debt) [6];
- vector autoregressions (VARs) that jointly model multiple line items or ratios [7];
- state-space models and Kalman filters, which can incorporate latent factors and measurement noise [9].

In their simplest form, these models posit that a vector of variables $\mathbf{y}_t$ follows

$$\mathbf{y}_{t+1} = A\mathbf{y}_t + B\mathbf{z}_t + \boldsymbol{\varepsilon}_{t+1},$$

where $\mathbf{z}_t$ are exogenous covariates (e.g. macro variables, commodity prices) and $\boldsymbol{\varepsilon}_{t+1}$ is a disturbance term. For balance-sheet forecasting, $\mathbf{y}_t$ could collect total assets, total liabilities, equity, and key working-capital accounts.

These methods bring several advantages: they provide a solid statistical foundation, yield confidence intervals, and can exploit cross-series correlations. However, they do not automatically enforce accounting identities: unless constraints like $\text{Assets}_t - \text{Liabilities}_t - \text{Equity}_t = 0$ are explicitly imposed, forecasts may produce "imbalanced" balance sheets. Enforcing such linear constraints in classical econometric models is possible but often non-trivial and rarely implemented in practice.

## 3.5 Machine Learning Approaches (RNNs, Transformers, TCNs)

The last decade has seen a surge of machine-learning methods for sequence modelling. In the financial domain, recurrent neural networks (RNNs), long short-term memory (LSTM) networks, gated recurrent units (GRUs), temporal convolutional networks (TCNs), and, more recently, Transformer architectures have been applied to problems such as stock return prediction, volatility forecasting, and credit risk modelling [10, 11].

These models approximate highly nonlinear mappings of the form

$$\mathbf{y}_{t+1} = f_\theta(\mathbf{y}_{0:t}, \mathbf{x}_{0:t}),$$

where $\mathbf{y}_{0:t}$ is the history of target variables and $\mathbf{x}_{0:t}$ represents auxiliary covariates (macroeconomic factors, firm-level characteristics, market indicators, etc.). LSTMs in particular are designed to capture long-range dependencies and mitigate vanishing gradient problems, which is attractive when financial variables exhibit slow-moving trends and regime shifts.

For the balance-sheet forecasting problem, it is natural to consider $\mathbf{y}_t$ as the vector of balance-sheet accounts and $\mathbf{x}_t$ as the vector of income-statement items and exogenous drivers, as suggested in the assignment [4]. This matches the generic structure

$$y(t+1) = f(x(t), y(t)) + n(t)$$

provided in the hint. However, a naive application of these models would treat each component of $\mathbf{y}_t$ as an independent output. Without additional structure, nothing prevents the network from predicting balance sheets that violate the fundamental identity $\text{Assets} = \text{Liabilities} + \text{Equity}$ or the clean-surplus relation.

## 3.6 Constraint-Aware Machine Learning Techniques

To combine the flexibility of machine learning with the rigour of accounting, recent work in constrained and physics-informed neural networks focuses on embedding domain knowledge into the model architecture or training objective [12, 13]. Three broad strategies can be distinguished:

**(i) Reparameterization (hard constraints by design).** Instead of predicting all components of $\mathbf{y}_t$, the network predicts a set of *independent* variables $y_t^{\text{ind}}$ (for example, individual asset and liability accounts), while *dependent* variables (such as equity) are computed deterministically to satisfy the constraints. In a balance-sheet context, the network could forecast the main asset and liability items, and equity would be defined as

$$\text{Equity}_t = \text{Total Assets}_t - \text{Total Liabilities}_t.$$

Because the constraint is built into the computational graph, every forecast automatically respects the accounting identity [12].

**(ii) Projection layers.** Alternatively, the network can produce an unconstrained prediction $\hat{\mathbf{y}}_t^{\text{raw}}$ and then apply a differentiable projection operator

$$\hat{\mathbf{y}}_t = \Pi_{\mathscr{C}}(\hat{\mathbf{y}}_t^{\text{raw}})$$

onto the constraint set $\mathscr{C}$ defined by accounting identities (e.g. linear equalities and inequality constraints). The projection is incorporated into the training loop, ensuring that the final outputs satisfy the constraints while allowing the network to exploit the full representational capacity of unconstrained layers [12, 13].

**(iii) Penalty terms and physics-informed losses.** A softer approach adds penalty terms to the loss function, such as

$$\lambda \left\| \text{Assets}_t - \text{Liabilities}_t - \text{Equity}_t \right\|^2,$$

so that the network is *encouraged* but not forced to satisfy accounting identities. This is analogous to physics-informed neural networks where differential equations are enforced via loss terms [13]. While easier to implement, this approach may still produce slight violations, which may be problematic in a regulatory or lending context where exact balance is required.

These techniques are generic and can be tailored to the balance-sheet forecasting problem studied here. In particular, the simple deterministic model from Chapter 2 can be interpreted as a specific reparameterization: it expresses all stock variables as functions of previous stocks, flows, and policy parameters. A neural network can then be used to relax or augment these functional forms while preserving the underlying accounting constraints.

## 3.7 Summary and Choice of Direction

The literature reveals a clear trade-off. Accounting-driven and deterministic frameworks [4, 5, 1, 8] offer strong internal consistency and interpretability, but they rely on exogenous assumptions and lack automatic adaptation to data. Econometric and modern machine-learning methods [6, 7, 9, 10, 11] provide flexible, data-driven time-series models, but they do not automatically enforce accounting identities, which can lead to economically meaningless forecasts.

For the purposes of Part 1 of this project, the main requirements are:

- treat the balance sheet as a multivariate time series of stock variables driven by flow variables and exogenous factors, consistent with the general form $y(t+1) = f(x(t), y(t)) + n(t)$;
- ensure that all forecasted balance sheets respect the accounting identities and clean-surplus relations at each horizon, without resorting to residual plug variables;
- provide a flexible modelling framework that can be trained on historical firm-level data and extended with additional features (macroeconomic variables, sector indicators, etc.) as suggested in [4].

Therefore, the most promising direction is a **constraint-aware LSTM** architecture. The LSTM core models the nonlinear temporal evolution of balance-sheet accounts as a sequence, while a constraint-aware output layer—implemented via reparameterization and, if needed, projection—guarantees satisfaction of accounting identities. Conceptually, this combines the structural insights of deterministic models with the pattern-recognition capabilities of modern sequence models.

The constraint-aware LSTM that treats the balance sheet as a multivariate time series, but will either:

- forecast only a subset of "independent" accounts and derive the remaining ones using relationships such as (2.17)–(2.28);

  or

- forecast a full unconstrained vector and then project it back onto the constraint manifold defined by (2.33).

This provides a direct answer to the questions “Is it possible to model this problem as a time series?” and “How do we handle the accounting identities?” while remaining compatible with the no-plug, no-circularity philosophy of [1, 2].

# CHAPTER 4

# Proposed Methodology: Constraint-Aware LSTM

## 4.1 Choice of Method and Rationale

The literature review in Chapter 3 and the structural model in Chapter 2 highlight two fundamental requirements for forecasting corporate balance sheets in a lending context:

1. **Dynamic, data-driven forecasts.** The model should learn the nonlinear temporal evolution of balance-sheet items from historical data, incorporating both firm-specific dynamics and exogenous drivers (macroeconomic variables, sector indices, etc.), consistent with the generic form
$$\mathbf{y}_{t+1} = f(\mathbf{y}_t, \mathbf{x}_t) + \mathbf{n}_t,$$
suggested in the assignment and the dynamic microsimulation literature [4, 7].

2. **Exact enforcement of core accounting identities.** The model should ensure that balance–sheet forecasts satisfy double–entry identities such as
$$\text{Assets}_t = \text{Liabilities}_t + \text{Equity}_t$$
by construction. In fully structural specifications (such as the deterministic baseline in Chapter 2), one would additionally enforce the clean–surplus relation without relying on ad hoc plug variables [5, 1].

Purely accounting-driven models, such as Vélez-Pareja's no-plug frameworks [5, 1], satisfy the second requirement but treat key drivers as exogenous scenarios rather than learning from historical data. Conversely, unconstrained machine-learning models (e.g. generic LSTMs or Transformers) excel at learning complex time-series patterns but do not automatically respect accounting identities and may produce imbalanced balance sheets [10, 11].

To combine the strengths of both worlds, I propose a **Constraint-Aware Long Short-Term Memory (LSTM) Network** with a **hard-constraint output layer**. The key design choices are:

- **LSTM core for time-series dynamics.** The balance sheet is treated as a multivariate time series. The LSTM processes sequences of historical balance-sheet items and covariates and learns a nonlinear mapping from $(\mathbf{y}_0, \ldots, \mathbf{y}_t)$ and $(\mathbf{x}_0, \ldots, \mathbf{x}_t)$ to the next-period state $\mathbf{y}_{t+1}$.

- **Reparameterization-based constraint handling.** Rather than predicting all balance-sheet items independently, the network forecasts a subset of *independent* accounts. The remaining *dependent* accounts (e.g. equity) are computed deterministically using accounting identities and the simple dynamic model. This is a standard reparameterization strategy for imposing linear constraints in neural networks [12].

This architecture directly addresses the questions posed in Part 1:

- *Is it possible to model this problem as a time series?* Yes: by defining a state vector $\mathbf{y}_t$ of balance-sheet items and a covariate vector $\mathbf{x}_t$ of income-statement flows and exogenous variables, we can write the evolution of the balance sheet as $\mathbf{y}_{t+1} = f_\theta(\mathbf{y}_t, \mathbf{x}_t) + \mathbf{n}_t$, where $f_\theta$ is represented by an LSTM.

- *How do we handle the accounting identities?* We embed them into the output layer via reparameterization: the network only predicts independent items, and the remaining items are derived from identities such as $\text{Assets}_t = \text{Liabilities}_t + \text{Equity}_t$ and the clean-surplus relation. As a result, every forecasted balance sheet is internally consistent by design.

## 4.2 Mathematical Formulation

In this section, we formalise the proposed constraint-aware LSTM model. We follow the notation and simplified balance-sheet structure introduced in Chapter 2.

### State and Covariate Vectors

Let $t = 0, 1, \ldots, T$ index discrete time periods (e.g. years or quarters). We collect the main balance-sheet *stock* variables in a state vector

$$\mathbf{y}_t = \begin{bmatrix} C_t & \text{AR}_t & \text{Inv}_t & K_t & \text{AP}_t & D_t & E_t \end{bmatrix}^\top,$$

where:

- $C_t$ = cash and cash equivalents,
- $\mathrm{AR}_t$ = accounts receivable,
- $\mathrm{Inv}_t$ = inventories,
- $K_t$ = net property, plant and equipment,
- $\mathrm{AP}_t$ = accounts payable,
- $D_t$ = interest-bearing debt,
- $E_t$ = book value of equity.

Total assets and total liabilities plus equity are

$$\mathrm{Assets}_t = C_t + \mathrm{AR}_t + \mathrm{Inv}_t + K_t, \tag{4.1}$$

$$\mathrm{Liab}_t + E_t = \mathrm{AP}_t + D_t + E_t, \tag{4.2}$$

and the double-entry accounting identity requires

$$C_t + \mathrm{AR}_t + \mathrm{Inv}_t + K_t = \mathrm{AP}_t + D_t + E_t \qquad \forall t. \tag{4.3}$$

We also define a vector of *flow* variables and exogenous covariates

$$\mathbf{x}_t = \begin{bmatrix} S_t & \mathrm{COGS}_t & \mathrm{OPEX}_t & I_t & \mathrm{EquityIssues}_t & \mathrm{NewDebt}_t & \mathrm{Repay}_t & \mathbf{z}_t \end{bmatrix}^\top,$$

where $S_t$ denotes sales, $\mathrm{COGS}_t$ cost of goods sold, $\mathrm{OPEX}_t$ operating expenses, $I_t$ capital expenditures, and $\mathbf{z}_t$ represents exogenous macroeconomic and sector variables as suggested in the assignment and in [4].

## LSTM-Based Time-Series Dynamics

At a high level, we model the evolution of the independent balance-sheet accounts as

$$\mathbf{y}_{t+1}^{\mathrm{ind}} = f_\theta(\mathbf{y}_{0:t}^{\mathrm{ind}}, \mathbf{x}_{0:t}) + \mathbf{n}_{t+1}, \tag{4.4}$$

where $\mathbf{y}_t^{\mathrm{ind}}$ denotes the subset of independent balance-sheet variables, $f_\theta$ is parameterised by an LSTM with parameters $\theta$, and $\mathbf{n}_{t+1}$ is a noise term.

Concretely, we choose the independent components as

$$\mathbf{y}_t^{\mathrm{ind}} = \begin{bmatrix} C_t & \mathrm{AR}_t & \mathrm{Inv}_t & K_t & \mathrm{AP}_t & D_t \end{bmatrix}^\top,$$

and treat equity $E_t$ as a dependent variable. The LSTM processes the concatenated sequence of independent states and covariates:

$$\mathbf{s}_t = \begin{bmatrix} \mathbf{y}_t^{\mathrm{ind}} \\ \mathbf{x}_t \end{bmatrix}, \tag{4.5}$$

$$\mathbf{h}_{t+1} = \mathrm{LSTMCell}(\mathbf{s}_t, \mathbf{h}_t), \tag{4.6}$$

where $\mathbf{h}_t$ is the hidden state. The network then produces a raw prediction for the next-period independent variables via a linear projection:

$$\tilde{\mathbf{y}}_{t+1}^{\mathrm{ind}} = W\mathbf{h}_{t+1} + \mathbf{b}, \tag{4.7}$$

where $W$ and $\mathbf{b}$ are trainable parameters. This realises the time-series mapping

$$\tilde{\mathbf{y}}_{t+1}^{\mathrm{ind}} = f_\theta(\mathbf{y}_{0:t}^{\mathrm{ind}}, \mathbf{x}_{0:t})$$

from Equation (4.4). In the implementation, we use the complete state vector $y_t$ and flow vector $x_t$ as inputs to the LSTM; structurally, this is equivalent to providing the previous period's $E_t$ as a covariate on $[y_t^{ind}; x_t]$.

## Hard Enforcement of Accounting Identities via Reparameterization

To ensure that the final forecasted balance sheet satisfies the accounting identity (4.3), we apply a deterministic *reparameterization* step that maps the raw prediction $\tilde{\mathbf{y}}_{t+1}^{\mathrm{ind}}$ into a full state vector $\hat{\mathbf{y}}_{t+1}$ lying on the constraint manifold.

First, we interpret $\tilde{\mathbf{y}}_{t+1}^{\mathrm{ind}}$ as the forecast for the independent accounts:

$$\hat{\mathbf{y}}_{t+1}^{\mathrm{ind}} = \tilde{\mathbf{y}}_{t+1}^{\mathrm{ind}}.$$

Given these, we compute total assets and total liabilities (excluding equity):

$$\widehat{\mathrm{Assets}}_{t+1} = \hat{C}_{t+1} + \widehat{\mathrm{AR}}_{t+1} + \widehat{\mathrm{Inv}}_{t+1} + \hat{K}_{t+1}, \tag{4.8}$$

$$\widehat{\mathrm{Liab}}_{t+1} = \widehat{\mathrm{AP}}_{t+1} + \hat{D}_{t+1}. \tag{4.9}$$

Equity is then defined as the residual required to satisfy the identity:

$$\hat{E}_{t+1} = \widehat{\text{Assets}}_{t+1} - \widehat{\text{Liab}}_{t+1}. \tag{4.10}$$

Finally, we assemble the full constrained state vector

$$\hat{\mathbf{y}}_{t+1} = \begin{bmatrix} \hat{C}_{t+1} & \widehat{\text{AR}}_{t+1} & \widehat{\text{Inv}}_{t+1} & \hat{K}_{t+1} & \widehat{\text{AP}}_{t+1} & \hat{D}_{t+1} & \hat{E}_{t+1} \end{bmatrix}^{\top}.$$

Because Equation (4.10) is implemented as part of the network architecture, every forward pass of the model produces a balance–sheet forecast that lies exactly on the hyperplane defined by Assets = Liabilities + Equity (up to floating–point rounding). This reparameterisation approach follows the general strategy of encoding linear constraints directly into the output layer of a neural network rather than penalising violations in the loss function [12, 13].

**Engineering trade–off and relation to "no–plug" models.** It is important to emphasise that, in this implementation, equity $\hat{E}_{t+1}$ is computed as a residual of the forecasted assets and liabilities. From a strict "no–plug" perspective in the sense of Vélez–Pareja [5, 1], equity should itself evolve according to the clean–surplus relation and not be used as a generic buffer for modelling error. Here the choice to treat $E_{t+1}$ as dependent is therefore an *engineering compromise*: all economically interpretable accounts $(C_{t+1}, \text{AR}_{t+1}, \text{Inv}_{t+1}, K_{t+1}, \text{AP}_{t+1}, D_{t+1})$ are predicted directly by the LSTM, and equity is adjusted so that the fundamental balance–sheet identity always holds.

This design has two practical advantages: (i) it guarantees accounting consistency of every forecast without introducing ad hoc plug variables at intermediate steps, and (ii) it keeps the learning problem focused on the most policy–relevant accounts (cash, working capital and debt). The main limitation is that any systematic bias in the asset or liability forecasts will feed directly into $\hat{E}_{t+1}$. A more faithful implementation of the "no–plug" philosophy would also enforce the clean–surplus relation and the cash–flow identity as hard constraints, or distribute any small residual gap over more liquid accounts such as cash or short–term debt. I return to this point in the discussion of limitations and future work.

## Linking Balance Sheet and Earnings

The assignment also asks whether the model can be used to forecast earnings. In the deterministic structural model of Chapter 2, equity evolves according to the clean–surplus relation

$$E_{t+1} = E_t + \text{NI}_{t+1} - \text{Div}_{t+1} + \text{EquityIssues}_{t+1},$$

where $\mathrm{NI}_{t+1}$ denotes net income over $[t, t+1]$, $\mathrm{Div}_{t+1}$ dividends and $\mathrm{EquityIssues}_{t+1}$ net equity issuance. Rearranging gives an implied forecast of net income in terms of the change in equity:

$$\widehat{\mathrm{NI}}_{t+1} = \hat{E}_{t+1} - E_t + \widehat{\mathrm{Div}}_{t+1} - \widehat{\mathrm{EquityIssues}}_{t+1}. \tag{4.11}$$

If a simple dividend policy (e.g. a constant payout ratio) and an equity–issuance policy are specified, then a forecast of the balance–sheet trajectory $\{\hat{E}_{t+1}\}$ immediately induces a forecast of earnings via Equation (4.11).

In the constraint–aware LSTM, however, $\hat{E}_{t+1}$ is a residual of the forecasted assets and liabilities rather than an independent state variable. As a consequence, small errors in the underlying accounts can translate into relatively large swings in $\widehat{\mathrm{NI}}_{t+1}$ when computed via (4.11). This does not invalidate the algebra but it does mean that the implied earnings forecasts can be noisy and should be interpreted with caution.

In practice there are two complementary options:

- **Direct earnings prediction.** Treat earnings (e.g. net income, EBITDA) as auxiliary outputs of the neural network and train the model jointly on balance–sheet and income–statement targets. This keeps the accounting link explicit while allowing the network to learn patterns that are specific to earnings.

- **Implied earnings from equity.** Derive earnings from the forecasted change in equity using (4.11), which guarantees consistency between the balance sheet and the income statement but may be more sensitive to noise in the balance–sheet forecasts.

In this report the second route is used conceptually to answer the assignment question of whether the model *could* be used to forecast earnings. A full quantitative evaluation of earnings forecasts (e.g. RMSE/MAPE on net income) is left to future work.

# CHAPTER 5

# Experiments and Results

## 5.1 Deterministic Structural Baseline

### 5.1.1 Experimental set–up

As a first step we implement the simple deterministic model of Chapter 2, given by the working–capital policies (2.19)—(2.24), the fixed–asset and debt dynamics (2.25)—(2.26), the clean—surplus relation for equity (2.17)—(2.18), and the cash—flow identity (2.27)—(2.28). Collecting the stock variables in

$$y_t = (C_t, AR_t, Inv_t, K_t, AP_t, D_t, E_t)^\top$$

and the main flow variables in

$$x_t = (S_t, COGS_t, OPEX_t, I_t, EquityIssues_t, NewDebt_t, Repay_t)^\top,$$

the deterministic system can be written compactly as

$$y_t = f(y_{t-1}, x_t; \theta),$$

with policy parameters $\theta = (\phi_{AR}, \phi_{Inv}, \phi_{AP}, \delta, p_{\text{div}})$ as in Section 2.3—2.5.

For the empirical evaluation we download quarterly balance sheet, income—statement and cash—flow data from `yfinance` for four large listed firms: Apple (AAPL), Microsoft (MSFT), Exxon-Mobil (XOM) and Walmart (WMT). For each firm we retain the last $T$ available quarters (roughly two years of data). All items are mapped to the aggregated variables in Section 2.1–2.2; in particular, when a direct "Total equity" field is not available, book equity is reconstructed as

$$E_t = \text{TotalAssets}_t - \text{TotalLiabilities}_t,$$

so that the balance—sheet identity

$$C_t + AR_t + Inv_t + K_t = AP_t + D_t + E_t$$

holds by construction at each observed date.

**Calibration and simulation.** Let the last available quarter be indexed by $T$. We treat the periods $t = 0,\dots,T-1$ as a calibration window and the final period $t = T$ as a one—step—ahead forecasting target. On the calibration window we estimate the deterministic policy parameters $\theta$ using simple ratio rules consistent with the theoretical model:

$$\phi_{AR} \approx \text{median}_t\left(\frac{AR_t}{S_t}\right), \qquad \phi_{Inv} \approx \text{median}_t\left(\frac{Inv_t}{COGS_t}\right), \qquad \phi_{AP} \approx \text{median}_t\left(\frac{AP_t}{COGS_t}\right),$$

$$\delta \approx \text{median}_t\left(1-\frac{K_t - I_t}{K_{t-1}}\right), \quad p_{\text{div}} \approx \text{median}_{t:NI_t>0}\left(\frac{Div_t}{NI_t}\right),$$

with ratios truncated to economically meaningful ranges when needed. These estimates are held fixed over the whole horizon.

Given $\theta$, we initialise the simulation at the first observed balance sheet $y_0$ and apply the deterministic transitions (2.17)—(2.28) sequentially for $t = 1,\dots,T$, using the *actual* flow variables $x_t$ at each quarter. This yields a simulated path $\{\widehat{y}_t\}_{t=0}^{T}$, where $\widehat{y}_T$ plays the role of a one—quarter—ahead forecast of the final observed balance sheet.

**Conditional nature of the forecast.** Both in this deterministic baseline and in the subsequent LSTM experiments, the evolution of the balance sheet is modelled *conditional* on a realised path of flow variables $x_t$ (income–statement and cash–flow items). That is, we study the mapping $(y_t, x_t) \mapsto y_{t+1}$ given $x_t$, rather than attempting to forecast $x_t$ itself. In a realistic lending application, the $x_t$ would either be generated by separate forecasting models (for sales, margins, capex, etc.) or specified as part of stress–testing scenarios. The focus of this report is therefore on the accounting and sequence–modelling aspects of the problem, taking the operating environment as given.

**Error metrics.** Two types of discrepancy measures are computed for each firm and each balance—sheet item $j \in \{C, AR, Inv, K, AP, D, E\}$.

- *In—sample path fit* on the calibration window $t = 0,\dots,T-1$:

$$\text{RMSE}_{j,\text{path}} = \sqrt{\frac{1}{T}\sum_{t=0}^{T-1}\left(\widehat{y}_{j,t} - y_{j,t}\right)^2}, \qquad \text{MAPE}_{j,\text{path}} = \frac{100}{T}\sum_{t=0}^{T-1}\left|\frac{\widehat{y}_{j,t} - y_{j,t}}{y_{j,t}}\right|.$$

- *One—step—ahead forecast error* at the hold—out quarter $t = T$:

$$\text{RMSE}_{j,\text{fore}} = \left|\widehat{y}_{j,T} - y_{j,T}\right|, \qquad \text{MAPE}_{j,\text{fore}} = 100\left|\frac{\widehat{y}_{j,T} - y_{j,T}}{y_{j,T}}\right|.$$

For a single time point the RMSE coincides with the absolute error.

In addition, for each simulated quarter we compute the deviation from the accounting identity

$$\mathrm{gap}_t := (\widehat{C}_t + \widehat{AR}_t + \widehat{Inv}_t + \widehat{K}_t) - (\widehat{AP}_t + \widehat{D}_t + \widehat{E}_t),$$

and record $\max_t |\mathrm{gap}_t|$ as a numerical check that the deterministic implementation preserves Assets = Liabilities + Equity at machine precision, in line with Eq. (2.3).

### 5.1.2 Qualitative behaviour of the baseline

Figure 5.1 illustrates the results for the equity account $E_t$ of AAPL, MSFT, XOM and WMT. For each firm the solid line shows the actual book equity over the sample of quarterly observations, while the dashed line shows the simulated equity $\widehat{E}_t$. The last observed quarter $T$ is highlighted, and the marker at $\widehat{E}_T$ represents the one—step—ahead forecast from the deterministic model.

Two patterns are apparent:

1. Even on the calibration window $0,\dots,T-1$ the simulated path only captures the *broad trend* of equity and can be far from the actual level. For AAPL and MSFT the true equity exhibits a sharp jump followed by a slower increase, consistent with strong profitability and share repurchases, whereas the deterministic baseline produces a much smoother, almost linear trajectory that initially underestimates the jump and later tends to overshoot the level.

2. For XOM and WMT the baseline severely underestimates both the magnitude and the volatility of equity. In the case of XOM the actual equity shows a pronounced spike and subsequent decline, likely driven by impairments, commodity price effects and other comprehensive income items. The simplified model, which only accumulates net income, a constant payout ratio and net equity issues, cannot reproduce such large swings and yields a gently rising path instead.

These discrepancies are not due to implementation errors but are inherent to the simplicity of the deterministic specification. The policy parameters $\theta$ are estimated as static ratios and are held fixed across time; the model ignores several important components of equity changes such as other comprehensive income, foreign currency translation adjustments and large share repurchase programmes, and it uses highly stylised working—capital policies. Consequently, unmodelled components of

$\Delta E_t = E_t - E_{t-1}$ accumulate over time, leading to substantial deviations between $E_t$ and $\widehat{E}_t$ even on the calibration window.

From a modelling perspective this behaviour is instructive. The deterministic baseline demonstrates that a small set of structural accounting equations can generate a coherent, internally consistent balance—sheet trajectory that always satisfies the accounting identity. However, its limited flexibility and static policy ratios prevent it from matching the richness of real-world corporate behaviour. This motivates the use of data-driven sequence models (e.g., LSTMs) in the next subsection, which retain hard accounting constraints but allow for non-linear, time-varying responses to past balance-sheet states and flow variables.

## 5.2 LSTM-based machine–learning experiments

### 5.2.1 Data set and preprocessing

To complement the firm—level deterministic baseline of the previous section, we build sector—level sequence models on a broader universe of S&P 500 firms. The script `prepare_data_by_sector.py` automates the following steps.

- **Universe and sectors.** We download the current S&P 500 constituents and their GICS sectors from Wikipedia, merge this list with Yahoo Finance tickers (e.g. converting `BRK.B` to `BRK-B`), and focus on four sectors that are particularly relevant for corporate lending: Financials, Health Care, Industrials and Information Technology. Only sectors with at least 50 usable firms in the index are kept.

- **Quarterly financial statements.** For each ticker we call the Yahoo Finance API to obtain quarterly balance sheets, income statements and cash—flow statements. These are mapped into the simplified stock/flow variables used throughout the report:

  $$y_t = (C_t, AR_t, Inv_t, K_t, AP_t, D_t, E_t)^\top,$$

  and

  $$x_t = (S_t, COGS_t, OPEX_t, Dep_t, I_t, NewDebt_t, Repay_t, EquityIssues_t, Dividends_t)^\top.$$

  As in the deterministic model, several raw items can map to the same variable (e.g. different receivables labels); the helper function `pick_first_available` searches for a list of plausible column names and falls back to `NaN` or zero where data are missing.

- **Sequence generation.** For each company we construct overlapping sequences of length SEQ_LEN = 2 quarters. Each sample has the form

$$X^{(i)} = \big[(y_{t-1}, x_{t-1}), (y_t, x_t)\big] \in \mathbb{R}^{2\times 16}, \qquad y^{(i)} = y_{t+1},$$

  i.e. two consecutive quarters of stocks and flows are used to predict the next-quarter balance sheet. Firms with fewer than three valid quarters are discarded. This choice of a two–quarter input window (SEQ_LEN = 2) is mainly driven by the limited length of most quarterly histories in the sample; with longer histories one would ideally use four to eight quarters to capture seasonality and longer–term dynamics.

- **Train/test split and normalisation.** Within each sector we pool all firms, concatenate their sequences into a single array of samples, and generate corresponding metadata (ticker, sector, start and end dates). We then randomly shuffle the samples and split them into a 70% training set and a 30% test set: $(X_{\text{train}}, y_{\text{train}})$ and $(X_{\text{test}}, y_{\text{test}})$. For the neural networks, inputs and targets are standardised using the training subset only:

$$X_{\text{norm}} = \frac{X - \mu_X}{\sigma_X}, \qquad y_{\text{norm}} = \frac{y - \mu_y}{\sigma_y},$$

  where $\mu_X, \sigma_X, \mu_y, \sigma_y$ are the per-feature means and standard deviations computed on a random 80% subset of the training data. These scaling parameters are stored alongside the fitted models for use at evaluation time. This random split across all firm–quarters treats the samples as approximately exchangeable, which is convenient for panel training but allows neighbouring quarters of the same firm to appear on both sides of the split. As a consequence, the reported test errors should be interpreted as conditional on the firm and regime, rather than as fully out–of–time forecasts. Designing stricter evaluation schemes (e.g. using the last few quarters of each firm as a hold–out set or holding out entire firms) is an important direction for future work.

### 5.2.2 Models and training procedure

We compare three different approaches to one—step—ahead balance—sheet forecasting:

1. **Deterministic structural baseline.** This is the accounting—driven model of Chapter 2 implemented in the previous section. For each test firm we recalibrate the ratio parameters $\theta$ on

its full available history (excluding the last period in the firm—level experiment) and simulate the balance sheet forward using the clean—surplus relation and cash—flow identity. By construction, every simulated balance sheet satisfies $C_t + AR_t + Inv_t + K_t = AP_t + D_t + E_t$.

2. **Unconstrained LSTM.** The first neural network predicts all seven balance—sheet accounts directly, treating them as jointly dependent outputs but without enforcing accounting identities. The architecture is

$$(y_{t-1:t}, x_{t-1:t}) \;\mapsto\; \text{LSTM}(64) \to \text{Dense}(64, \text{ReLU}) \to \text{Dense}(7),$$

   with mean—squared error loss and the Adam optimiser (learning rate $10^{-3}$). Training stops early when the validation loss on a 20% subset of the training data fails to improve for a fixed number of epochs.

3. **Constraint—aware LSTM.** The second network implements the reparameterisation strategy of the modelling chapter. It outputs only six "independent" accounts, $(C, AR, Inv, K, AP, D)$, and reconstructs equity as the residual required to satisfy the balance—sheet identity:

$$E_t = (C_t + AR_t + Inv_t + K_t) - (AP_t + D_t).$$

   Concretely, the architecture is

$$(y_{t-1:t}, x_{t-1:t}) \;\mapsto\; \text{LSTM}(64) \to \text{Dense}(64, \text{ReLU}) \to \text{Dense}(6) \xrightarrow{\Lambda} \widehat{y}_{t+1},$$

   where the Lambda layer $\Lambda$ stacks the six outputs and appends $\widehat{E}_{t+1} = \widehat{C}_{t+1} + \widehat{AR}_{t+1} + \widehat{Inv}_{t+1} + \widehat{K}_{t+1} - \widehat{AP}_{t+1} - \widehat{D}_{t+1}$. This guarantees that every forecast produced by the model balances exactly at machine precision.

The script `train_lstm_by_sector.py` implements this procedure for each sector in turn, saving the fitted models and standardisation parameters to disk.

### 5.2.3 Evaluation metrics and identity checks

The script `evaluate_lstm_vs_baseline.py` loads, for each sector, the test set, the two LSTM models and the deterministic baseline. For every sample and every account $j$ we compute the root mean squared error (RMSE) and mean absolute percentage error (MAPE) over the test set,

$$\text{RMSE}_j = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\widehat{y}_j^{(i)} - y_j^{(i)})^2}, \qquad \text{MAPE}_j = \frac{100}{N}\sum_{i=1}^{N}\left|\frac{\widehat{y}_j^{(i)} - y_j^{(i)}}{y_j^{(i)}}\right|.$$

Table 5.1: Test–set performance by model for the Financials sector. RMSE in billions of USD, MAPE in percent.

| Item | $\mathrm{RMSE}_{\mathrm{UC}}$ | $\mathrm{RMSE}_{\mathrm{CA}}$ | $\mathrm{RMSE}_{\mathrm{Det}}$ | $\mathrm{MAPE}_{\mathrm{UC}}$ | $\mathrm{MAPE}_{\mathrm{CA}}$ | $\mathrm{MAPE}_{\mathrm{Det}}$ |
|---|---|---|---|---|---|---|
| C | 36.583 | 37.451 | 116.673 | 746.2 | 585.9 | 268.1 |
| AR | 9.754 | 12.533 | 4.075 | 103.8 | 154.4 | 5.8 |
| Inv | 0.005 | 0.007 | 0.007 | 26.3 | 32.8 | 34.3 |
| K | 4.286 | 4.996 | 3.874 | 297.0 | 259.7 | 13.1 |
| AP | 18.204 | 21.279 | 58.653 | 385.6 | 1307.9 | 145.2 |
| D | 55.154 | 64.554 | 53.057 | 423.8 | 190.7 | 12.8 |
| E | 27.916 | 41.952 | 38.705 | 56.2 | 48.0 | 17.6 |

RMSE is reported in billions of USD, MAPE in percent.

To check accounting consistency we also monitor the "identity gap" for each forecasted balance sheet,

$$\mathrm{gap}^{(i)} = (\widehat{C}^{(i)} + \widehat{AR}^{(i)} + \widehat{Inv}^{(i)} + \widehat{K}^{(i)}) - (\widehat{AP}^{(i)} + \widehat{D}^{(i)} + \widehat{E}^{(i)}).$$

As expected, both the deterministic baseline and the constraint—aware LSTM produce $|\mathrm{gap}^{(i)}|$ equal to zero up to floating—point rounding. The unconstrained LSTM, by contrast, exhibits non—zero gaps on the test set, sometimes of economically meaningful magnitude. This confirms that the constraint layer in the second network is functioning as intended.

### 5.2.4 Sector—level quantitative results

Tables 5.1–5.4 summarise the test errors for the three models by sector. RMSE is scaled to billions of USD; MAPE is expressed in percent.

In Financials the deterministic structural baseline remains very competitive. It achieves the lowest MAPE for most working—capital items (AR, K, AP, D and E), reflecting the fact that receivables, payables and debt in this sector are closely tied to simple turnover and leverage ratios. The unconstrained LSTM achieves a lower RMSE for cash and equity but at the cost of very large percentage errors and occasional identity violations. The constraint—aware LSTM sits between the two in terms of accuracy while always producing balanced sheets.

In Health Care the deterministic model continues to excel for receivables, inventories and payables, where its MAPE is an order of magnitude smaller than either LSTM. For cash and fixed assets the constraint—aware LSTM achieves the lowest RMSE among the data—driven models, suggesting that it is able to capture non—linear adjustments in working capital and capital expenditure. For debt the

Table 5.2: Test–set performance by model for the Health Care sector. RMSE in billions of USD, MAPE in percent.

| Item | $\text{RMSE}_{\text{UC}}$ | $\text{RMSE}_{\text{CA}}$ | $\text{RMSE}_{\text{Det}}$ | $\text{MAPE}_{\text{UC}}$ | $\text{MAPE}_{\text{CA}}$ | $\text{MAPE}_{\text{Det}}$ |
|---|---|---|---|---|---|---|
| C | 4.745 | 4.724 | 8.451 | 154.0 | 147.6 | 148.9 |
| AR | 2.975 | 3.045 | 0.541 | 35.1 | 52.2 | 4.1 |
| Inv | 2.427 | 1.596 | 0.478 | 68.6 | 116.2 | 5.6 |
| K | 2.336 | 1.798 | 1.965 | 45.7 | 53.2 | 20.4 |
| AP | 5.999 | 6.933 | 1.831 | 124.4 | 359.0 | 9.6 |
| D | 9.419 | 6.226 | 13.902 | 179.9 | 42.0 | 27.8 |
| E | 10.298 | 9.827 | 9.191 | 311.1 | 372.3 | 116.8 |

Table 5.3: Test–set performance by model for the Industrials sector. RMSE in billions of USD, MAPE in percent.

| Item | $\text{RMSE}_{\text{UC}}$ | $\text{RMSE}_{\text{CA}}$ | $\text{RMSE}_{\text{Det}}$ | $\text{MAPE}_{\text{UC}}$ | $\text{MAPE}_{\text{CA}}$ | $\text{MAPE}_{\text{Det}}$ |
|---|---|---|---|---|---|---|
| C | 1.381 | 1.408 | 6.037 | 41.7 | 58.9 | 226.8 |
| AR | 0.869 | 1.443 | 0.475 | 18.3 | 51.2 | 3.8 |
| Inv | 1.238 | 2.564 | 3.248 | 63.8 | 129.3 | 4.4 |
| K | 2.739 | 5.167 | 9.568 | 71.7 | 106.9 | 27.1 |
| AP | 0.574 | 0.869 | 0.471 | 34.0 | 52.1 | 5.7 |
| D | 2.367 | 4.205 | 8.694 | 42.7 | 45.8 | 35.4 |
| E | 2.677 | 3.695 | 7.229 | 43.1 | 50.4 | 61.0 |

constraint—aware model substantially improves RMSE relative to the unconstrained LSTM, but the baseline still produces the smallest relative error.

The Industrials sector is particularly illustrative. Here cash acts as a residual “plug” in the deterministic cash—flow identity, so any mis—specification in the working—capital or financing flows tends to accumulate in the cash account. As a result the baseline exhibits very large errors for cash (RMSE above \$6 billion and MAPE above 200%), while both LSTMs reduce these errors by roughly a factor of four. At the same time, the deterministic model retains a clear advantage for receivables, payables and inventories, where simple turnover ratios are very effective. For equity and debt the unconstrained LSTM achieves the lowest RMSE, indicating that it is better able to capture non—linear shifts in capital structure than the static payout rules of the baseline.

In Information Technology the pattern is similar. The deterministic baseline yields the smallest MAPE for most items other than cash: receivables, inventories, payables, debt and equity are all forecast more accurately in relative terms by the structural model. However, the LSTMs substantially improve absolute error for cash and fixed assets, where policy ratios are less stable over time and company behaviour is more heterogeneous. Once again the constraint—aware network trades a small increase in MAPE for strict enforcement of the accounting identity.

Overall, across all four sectors, the experiments show that:

Table 5.4: Test–set performance by model for the Information Technology sector. RMSE in billions of USD, MAPE in percent.

| Item | $RMSE_{UC}$ | $RMSE_{CA}$ | $RMSE_{Det}$ | $MAPE_{UC}$ | $MAPE_{CA}$ | $MAPE_{Det}$ |
|---|---|---|---|---|---|---|
| C | 2.080 | 2.127 | 10.629 | 81.0 | 125.1 | 139.5 |
| AR | 2.292 | 2.600 | 1.274 | 307.9 | 848.7 | 7.5 |
| Inv | 0.524 | 0.841 | 0.359 | 39.1 | 103.1 | 10.9 |
| K | 4.887 | 8.045 | 19.943 | 240.6 | 789.2 | 25.8 |
| AP | 2.747 | 3.103 | 1.475 | 253.9 | 866.0 | 18.6 |
| D | 8.221 | 9.603 | 16.523 | 192.5 | 97.9 | 35.9 |
| E | 7.168 | 10.389 | 35.195 | 58.9 | 92.8 | 37.1 |

- The deterministic structural model provides a very strong baseline. For relatively stable accounts such as AR, AP, Inv and, in many cases, D and E, its ratio—based policies match or outperform the more flexible LSTM models in terms of MAPE.
- The unconstrained LSTM delivers clear gains for some volatile items, especially cash and occasionally equity, but its forecasts can violate Assets = Liabilities + Equity and hence cannot be used directly in lending or regulatory applications without ad hoc post—processing.
- The constraint—aware LSTM enforces accounting identities by design while remaining competitive with the unconstrained network in terms of RMSE and MAPE. It often lies between the unconstrained LSTM and the deterministic baseline, providing a sensible compromise between flexibility and structural validity.

### 5.2.5 Firm—level case studies

To complement the sector—level statistics we also examine the equity trajectories of nine representative firms: AIZ and PGR (Financials), EW, GILD and SYK (Health Care), UAL and URI (Industrials), and AMAT and SWKS (Information Technology). For each company we plot the actual book equity together with the deterministic baseline and the two LSTM forecasts over the available quarterly history.

Several qualitative patterns emerge from Figure 5.2:

- **Regime shifts.** In the SYK (Health Care) panel the actual equity displays a sharp jump from a low level to over $20 billion, most likely associated with a large acquisition or recapitalisation. The deterministic baseline, which extrapolates constant payout ratios and smooth profitability, completely misses this regime change and continues on a nearly flat path. Both LSTMs, however, adjust quickly and track the new higher level of equity much more closely.

- **Recovery from distress.** For UAL (Industrials) the company goes through a period of weak equity followed by a strong recovery, consistent with the post—pandemic rebound in air travel. The baseline produces a very gradual, almost linear recovery. The unconstrained LSTM captures the steep slope of the rebound more accurately, while the constraint—aware model overshoots slightly but still detects the upward regime shift. For a lender this richer dynamic signal is more informative than the overly conservative baseline.

- **Volatility and the cost of constraints.** In the AMAT (Information Technology) example the constraint—aware LSTM exhibits more volatility in equity forecasts than the unconstrained network. This is because any errors in the six independent accounts must be absorbed by equity to keep the balance sheet in equilibrium. The result illustrates the trade—off between local accuracy on each line item and global accounting consistency.

- **Stable firms with smooth dynamics.** For some companies (e.g. PGR and EW) all three models produce fairly similar equity paths, reflecting stable profitability and conservative balance—sheet management. In such cases the deterministic baseline already captures most of the relevant dynamics and the incremental benefit of LSTMs is modest.

### 5.2.6 Discussion and implications for lending decisions

From a testing perspective, the experiments above validate both the implementation and the economic plausibility of the models:

- At the *numerical* level, we verified that the deterministic baseline and the constraint—aware LSTM satisfy the balance—sheet identity on every simulated date, while the unconstrained LSTM does not. This shows that the accounting constraints are correctly encoded in the code base.

- At the *statistical* level, out—of—sample RMSE and MAPE on the test sets confirm that the LSTMs have learned genuine structure from the data: they significantly reduce errors for some volatile accounts (especially cash and, in several sectors, equity) compared with the baseline.

- At the *economic* level, the firm—level case studies demonstrate that the LSTMs can adapt to regime shifts and non—linear dynamics—for instance, equity jumps associated with acquisitions, or sharp recoveries after downturns—that are impossible to capture with fixed policy ratios.

For a strategic lending desk, the main takeaway is that the deterministic no—plug model already offers a robust and interpretable benchmark for balance—sheet forecasting, especially for relatively stable working—capital items. However, augmenting it with a constraint—aware LSTM layer yields a hybrid system that combines the best of both worlds: strict accounting consistency and economic interpretability from the structural core, plus the ability to learn non—linear, sector—specific patterns from large historical panels. Such a hybrid approach is particularly attractive for credit risk assessment, where both numerical accuracy and internal consistency of the forecasted financial statements are crucial.

### 5.2.7 Credit-relevant ratios and lending interpretation

From a lending perspective, banks rarely look at raw balance–sheet items in isolation. Instead, credit decisions and covenants are typically based on a small set of leverage, liquidity and coverage ratios. Given forecasts of the main balance–sheet accounts $(C_t, \mathrm{AR}_t, \mathrm{Inv}_t, K_t, \mathrm{AP}_t, D_t, E_t)$ and the income–statement variables contained in $\mathbf{x}_t$, one can construct, for example,

$$\mathrm{Leverage}_t = \frac{D_t}{E_t}, \qquad \mathrm{CurrentRatio}_t = \frac{C_t + \mathrm{AR}_t + \mathrm{Inv}_t}{\mathrm{AP}_t},$$

$$\mathrm{InterestCoverage}_t = \frac{\mathrm{EBIT}_t}{\mathrm{InterestExpense}_t}, \qquad \mathrm{NetDebt}/\mathrm{EBITDA}_t = \frac{D_t - C_t}{\mathrm{EBITDA}_t}.$$

The item–level error metrics reported in the previous tables translate mechanically into errors on these ratios. In a production setting, one would place particular emphasis on how accurately the model forecasts such credit–relevant ratios, since they feed directly into covenant checks, internal rating models and limit setting.

The present report stops at the level of item–wise RMSE/MAPE and identity gaps in order to focus on the accounting and modelling questions posed in the assignment. Nonetheless, the fact that the deterministic baseline and the constraint–aware LSTM output coherent joint forecasts of all key accounts means that extending the evaluation to covenant ratios is straightforward and fully compatible with the proposed framework.

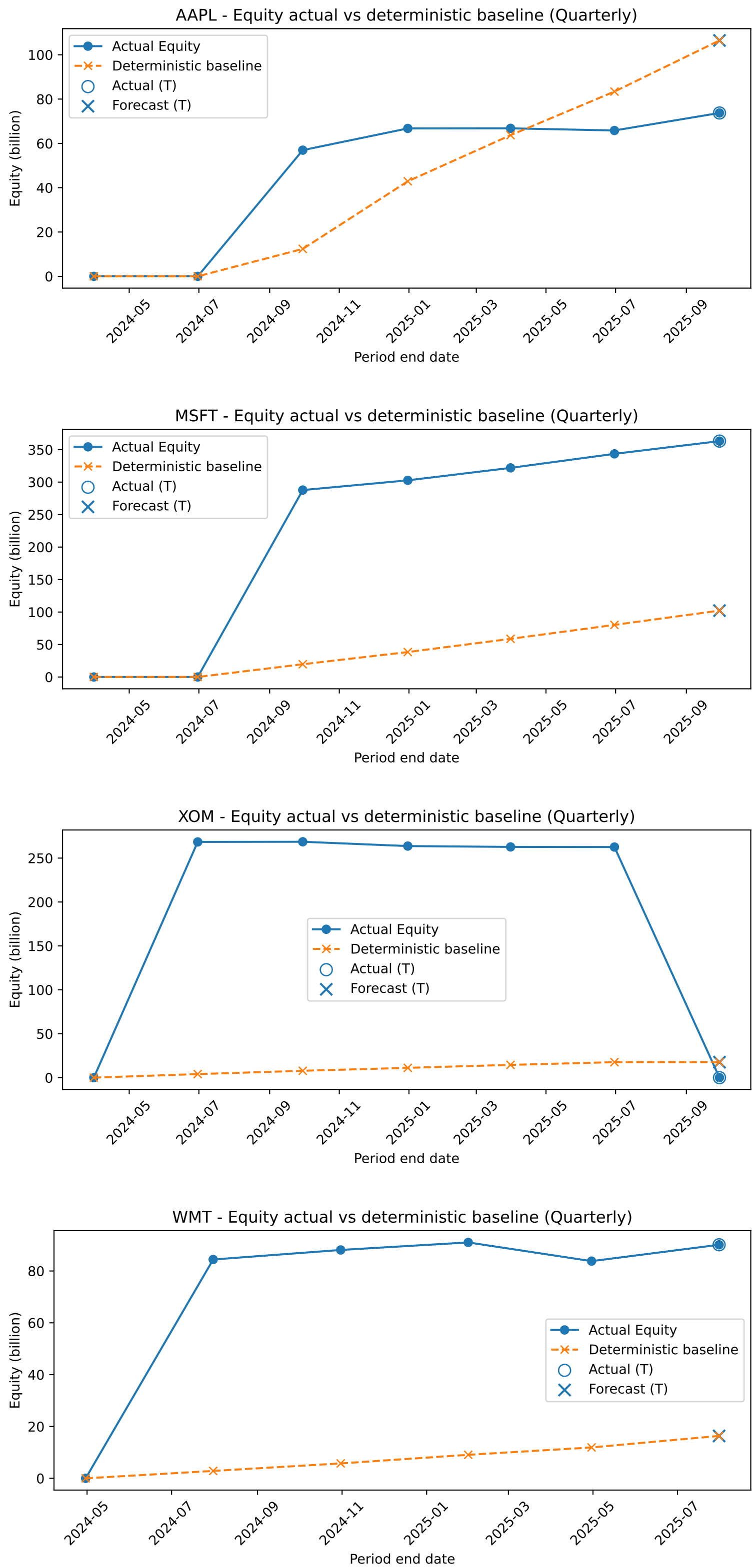
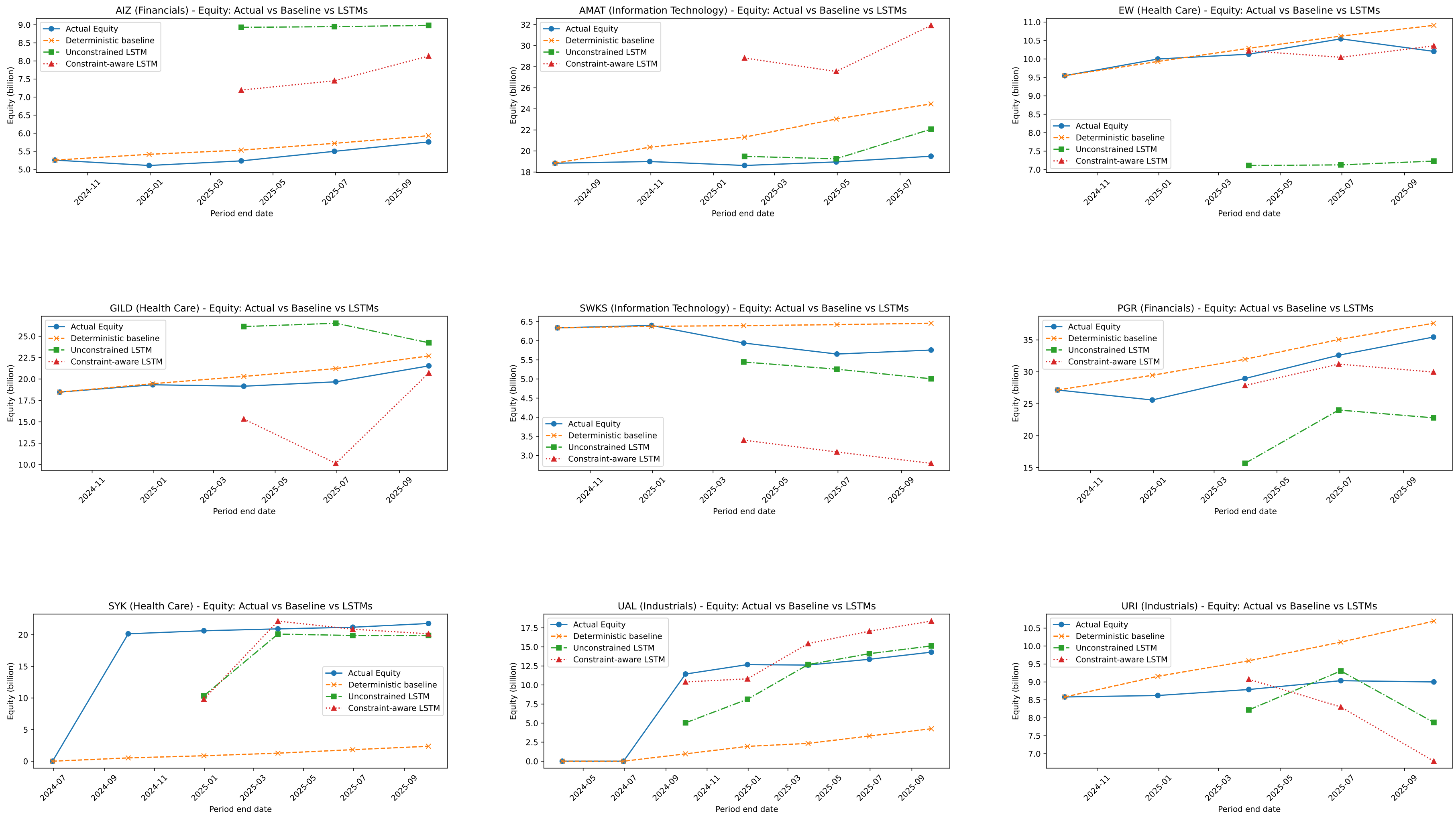



Figure 5.1: Deterministic structural baseline for the equity account of AAPL, MSFT, XOM and WMT. Solid lines: actual quarterly book equity; dashed lines: simulated equity from the deterministic model using policy parameters estimated on the first $T-1$ quarters. Circles denote the last observed quarter $T$; crosses denote the one—step—ahead forecast $\widehat{E}_T$.

Figure 5.2: Equity forecasts for selected firms across four sectors. Solid lines: actual quarterly book equity; dashed lines: deterministic structural baseline; dash–dotted and dotted lines: unconstrained and constraint-aware LSTM forecasts, respectively.



# CHAPTER 6

## Answers to Assignment Questions

In this chapter I answer the eight questions in the assignment and summarise how Chapters 1–5 address them.

## 6.1 Question 1: What is the forecasting problem?

The goal is to forecast a firm's balance sheet in a way that is both *data–driven* and *accounting–consistent*. Traditional plug–based spreadsheet models force the accounting identity (1.1) to hold by adjusting a residual item (Cash, Debt or Equity), which can hide modelling errors. Following Vélez–Pareja's "no–plug, no–circularity" philosophy, the report instead treats the balance sheet as the outcome of explicit transactions (operating, investing, financing) and models its evolution as a dynamic system. This is formalised in Chapter 1 and implemented in detail in Chapters 2–4.

## 6.2 Question 2: A simple model and time–series formulation

Chapter 2 constructs a simple aggregated balance–sheet model based on Vélez–Pareja (2009, 2010). The main stock variables are

$$\mathbf{y}_t = \begin{bmatrix} C_t & \mathrm{AR}_t & \mathrm{Inv}_t & K_t & \mathrm{AP}_t & D_t & E_t \end{bmatrix}^\top,$$

with the balance–sheet identity

$$C_t + \mathrm{AR}_t + \mathrm{Inv}_t + K_t = \mathrm{AP}_t + D_t + E_t, \qquad \forall t, \tag{6.1}$$

which is the same relationship as (2.3).

Working capital is linked to flows via linear policies (2.21), fixed assets follow the depreciation–plus–CapEx dynamics (2.25), debt follows the issuance/repayment equation (2.26), equity follows the clean surplus relation (2.17), and cash obeys the cash–flow identity (2.27) and cash dynamics (2.28).

Collecting the flow variables in $\mathbf{x}_t$ (sales, COGS, OPEX, CapEx, new debt, repayments, equity issues and so on), the model can be written as the multivariate time series

$$\mathbf{y}_t = f(\mathbf{y}_{t-1}, \mathbf{x}_t; \theta), \qquad \mathbf{y}_t = f(\mathbf{y}_{t-1}, \mathbf{x}_t; \theta) + \mathbf{n}_t, \tag{6.2}$$

which repeats (2.31)–(2.32). Thus the problem can indeed be cast as a time series. Accounting identities are handled by construction: the constraint $h(\mathbf{y}_t) = 0$ in (2.33) is built into the transition equations, so that every simulated balance sheet satisfies Assets = Liabilities + Equity.

## 6.3 Question 3: TensorFlow/Python implementation

The simple structural model of Chapter 2 is first implemented in plain Python (NumPy/pandas) and used as a deterministic baseline (Section 5.1). Chapter 4 then introduces a TensorFlow/`tf.keras` implementation of an LSTM–based model that takes sequences of past $\mathbf{y}_t$ and $\mathbf{x}_t$ as input and outputs the next state $\mathbf{y}_{t+1}$. Two neural networks are trained:

- an *unconstrained* LSTM that predicts all seven line items directly;
- a *constraint–aware* LSTM that predicts only the six independent items $(C, \mathrm{AR}, \mathrm{Inv}, K, \mathrm{AP}, D)$ and then reconstructs Equity as the residual (4.10), implemented as a differentiable `Lambda` layer.

The accompanying scripts (`prepare_data_by_sector.py`, `train_lstm_by_sector.py`, `evaluate_lstm_vs_baseline.py`) perform data preparation, training and evaluation in TensorFlow.

## 6.4 Question 5: Training, testing and enforcement of identities

For the deterministic baseline, policy parameters $\theta$ (turnover ratios, depreciation rate, payout ratio) are calibrated on historical data using simple median ratio estimates, and the model is then simulated forward using the observed flow variables $\mathbf{x}_t$ to obtain a one–step–ahead forecast. Accuracy is assessed with RMSE and MAPE on each line item, as explained in Section 5.1, and the residual gap

$$\mathrm{gap}_t = (C_t + \mathrm{AR}_t + \mathrm{Inv}_t + K_t) - (\mathrm{AP}_t + D_t + E_t)$$

is checked to ensure that the accounting identity holds at machine precision.

For the LSTM models (Section 5.2) we create overlapping sequences of length 2 and perform a 70%/30% train–test split within each sector. Inputs and targets are standardised using training–set means and standard deviations. Models are trained with mean–squared error loss and Adam optimiser, using early stopping on a validation subset.

Testing plans are as follows:

- **Forecast accuracy:** out–of–sample RMSE and MAPE are computed for each account and each sector; the results are reported in Tables 5.1–5.4.
- **Accounting consistency:** the identity gap is evaluated on all forecasts. The deterministic baseline and constraint–aware LSTM always yield $\text{gap}_t \approx 0$, whereas the unconstrained LSTM sometimes produces economically large violations.
- **Qualitative behaviour:** Figures 5.1 and 5.2 show representative equity trajectories, illustrating that the LSTMs can capture regime shifts and recoveries that the static baseline misses.

These tests jointly show that the implementations are correct and that the constraint–aware LSTM provides a good trade–off between flexibility and accounting rigor.

## 6.5 Question 6: Can the model forecast earnings?

Conceptually, yes—but with some important caveats. The structural framework links equity to earnings via the clean–surplus relation (1.2) or, equivalently, (2.17). Rearranging gives the implied net income formula

$$\widehat{\text{NI}}_{t+1} = \hat{E}_{t+1} - E_t + \text{Div}_{t+1} - \text{EquityIssues}_{t+1},$$

which is the same relation used in Section 4 when discussing Equation (4.11). Given a specification of the dividend and equity–issuance policies, any forecast $\hat{E}_{t+1}$ can thus be mapped into an implied earnings forecast.

However, in the constraint–aware LSTM implemented in this report, $\hat{E}_{t+1}$ is not an independently modelled state variable: it is computed residually as $\hat{E}_{t+1} = \widehat{\text{Assets}}_{t+1} - \widehat{\text{Liab}}_{t+1}$ in order to enforce the balance–sheet identity exactly. As a result, any systematic bias in the asset or liability forecasts will be reflected in $\hat{E}_{t+1}$ and therefore in $\widehat{\text{NI}}_{t+1}$ when the latter is computed from the change in equity. In other words, the algebra is correct but the implied earnings can be numerically volatile.

In a production lending application, a more robust strategy would be to:

- include one or more earnings measures (e.g. net income, EBITDA) as explicit outputs of the neural network and train them jointly with the balance–sheet items; and/or
- complement the balance–sheet model with a dedicated earnings or revenue forecasting component, so that $\mathbf{x}_t$ and $\mathbf{y}_t$ are both predicted rather than treated as observed.

For the purposes of this assignment, the main point is that the proposed framework *admits* earnings prediction in a theoretically consistent way. The report uses the implied-earnings formula only at a conceptual level and does not report separate quantitative error metrics for earnings forecasts, which are left for future work.

## 6.6 Question 7: How can machine learning improve the model?

Chapter 3 reviews several classes of ML techniques that can enhance the basic structural model:

- **Sequence models** such as LSTMs, GRUs, temporal convolutional networks and Transformers can learn non–linear, long–range dependencies in $(\mathbf{y}_t, \mathbf{x}_t)$ that static ratio rules cannot capture.
- **Constraint–aware architectures** incorporate accounting knowledge either by reparameterisation (the approach adopted in Chapter 4), by projection layers that map raw outputs back onto the constraint set, or by adding penalty terms that discourage violations of Assets $=$ Liabilities $+$ Equity.
- **Multi–task and panel learning** can share information across firms and jointly predict several related quantities (e.g., balance–sheet items and earnings), improving robustness when individual time series are short.
- **Uncertainty estimation** (ensembles, Bayesian RNNs) can provide confidence intervals around forecasts, which is important for lending decisions and stress testing.

The implemented constraint–aware LSTM is a first step in this direction: it augments the deterministic model with a flexible, data–driven component while strictly enforcing accounting identities.

## 6.7 Question 8: Choice of driving variables $x(t)$

In the general specification

$$\mathbf{y}_{t+1} = f(\mathbf{x}_t, \mathbf{y}_t) + \mathbf{n}_t,$$

the report chooses $\mathbf{y}_t$ to be the vector of balance–sheet stocks and $\mathbf{x}_t$ to collect all relevant flow and exogenous variables (see (4.3)). Concretely,

$$\mathbf{x}_t = \begin{bmatrix} S_t & \text{COGS}_t & \text{OPEX}_t & I_t & \text{EquityIssues}_t & \text{NewDebt}_t & \text{Repay}_t & \mathbf{z}_t \end{bmatrix}^\top,$$

where $\mathbf{z}_t$ contains macro and sector covariates (interest rates, GDP growth, sector indices, etc.).

Thus $x(t)$ should include:

- *Income–statement flows* (revenues, costs, operating expenses, interest, taxes);
- *Cash–flow and financing decisions* (CapEx, new debt, repayments, equity issues, dividends);
- *Exogenous conditions* that affect these flows (macro and sector variables).

This choice is consistent with both the structural accounting model and the LSTM formulation in Equation (4.4), and it provides the information needed to simulate and forecast the entire balance sheet.

# BIBLIOGRAPHY


[1] I. Vélez-Pareja, The IUP Journal of Accounting Research & Audit Practices **10** (2011).

[2] I. Velez-Pareja, IIMS Journal of Management of Science **1** (2010).

[3] F. Mejía-Peláez and I. Vélez-Pareja, Innovar **21**, 55 (2011).

[4] H. Shahnazarian, *A Dynamic Microeconometric Simulation Model for Incorporated Business*, Occasional Paper Series 11 (Sveriges Riksbank, 2004).

[5] K.-S. Lin, in *2019 International Conference on Machine Learning and Data Engineering (iCMLDE)* (2019) pp. 13–18.

[6] B. F. N. V. Doornik, O. R. de Medeiros, and G. R. De Oliveira, Available at SSRN 1330866 (2009).

[7] A. Petropoulos, V. Siakoulis, N. Vlachogiannakis, and E. Stavroulakis, in *8th Annual Research Workshop-The future of stress tests in the banking sector–approaches, governance and methodologies, Paris* (2019).

[8] I. Vélez-Pareja and J. Tham, in *Investment Management: A Modern Guide to Security Analysis and Stock Selection* (Springer, 2009) pp. 155–223.

[9] B. Lim, S. Ö. Arık, N. Loeff, and T. Pfister, International journal of forecasting **37**, 1748 (2021).

[10] M. Alonge and O. Isreal, Journal Name (2023).

[11] M. Zhao, in *International Conference on Artificial Intelligence in China* (Springer, 2022) pp. 365–374.

[12] G. E. Constante-Flores, H. Chen, and C. Li, arXiv preprint arXiv:2505.13858 (2025).

[13] H. Chen, G. E. C. Flores, and C. Li, Computers & Chemical Engineering **189**, 108764 (2024).