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

Guide to the Report: Roadmap to Solutions

This report provides an end-to-end solution to the *JP Morgan MLCOE TSRL 2026 Internship Question 1*. The structure follows the natural progression of the assignment:

Part 1: Modeling of a Balance Sheet

Assignment Question	**Where It Is Addressed in This Report**
Part 1 (1): Literature, motivation, and "no-plug" balance-sheet modeling with identities	**Chapter 2**: problem setup and accounting identities; overview of the "no-plug" philosophy and why balance-sheet fields are not independent.
Part 1 (2): Simple balance-sheet model; governing equations; time-series interpretation; how identities are enforced	**Chapter 3** (Secs. 2.1–2.2): deterministic transition rules written as a recursive simulation $y_{t+1} = f(x_t, y_t)$; identities enforced by construction (rather than post-hoc plugging).
Part 1 (3): Implementation in Python/TensorFlow	**Chapter 3** (Secs. 2.1–2.2): baseline implementation of the simulator and objective; **Chapter 4**: constrained ML parameterization in TensorFlow.
Part 1 (4): Data acquisition (Yahoo Finance / `yfinance`)	**Chapters 3 and 4**: statement collection and preprocessing pipeline (balance sheet + income statement), with practical notes for multi-firm experiments.
Part 1 (5): Company selection; training strategy; evaluation; identity checking	**Chapter 3** (Secs. 2.2–2.3): calibration, train/test splits, and diagnostic checks for accounting consistency; **Chapter 4** (Secs. 3.8–3.9): ML training/evaluation protocols and forecasting metrics under constraints.
Part 1 (6): Can the model forecast earnings?	**Chapter 3** (Sec. 2.4) and **Chapter 4**: earnings are forecasted consistently as model-implied quantities (coupled to the simulated balance-sheet dynamics), rather than treated as unconstrained independent targets.
Part 1 (7): ML techniques to improve forecasting	**Chapter 4**: constrained deep forecasting (e.g., time-varying parameters / feature-driven dynamics) while maintaining strict accounting identities; discussion of extensions (regularization, probabilistic noise, multi-task learning).
Part 1 (8): Simulation/prediction view; what are $x(t)$ and $n(t)$?	**Chapters 3 and 4**: the report adopts the simulation viewpoint explicitly via $y_{t+1} = f(x_t, y_t) + n_t$, where y_t is the balance-sheet state and x_t contains economically meaningful drivers (income-statement signals, growth/efficiency proxies, and firm covariates); n_t is handled via residual modeling and/or probabilistic training objectives.

Table 1.1: Mapping of Part 1 Questions to Chapters and Sections

Part 2: LLM Applications and Financial Analysis

Assignment Question	**Where It Is Addressed in This Report**
Part 2 (a): Choose an LLM for financial statement analysis	**Chapter 5**: LLM selection rationale, prompting design, and analysis scope for statement understanding and forecasting support.
Part 2 (b): LLM vs. constrained model on balance-sheet forecasting	**Chapter 5**: empirical comparison between (i) the structural simulator, (ii) the constrained ML model, and (iii) the LLM-based approach on the same dataset collected in Part 1.
Part 2 (c): Ensemble of Part 1 model and LLM	**Chapter 5**: hybrid/ensemble strategies (e.g., LLM as feature generator or regime tagger; constrained model as identity-preserving backbone; mixture or residual correction).
Part 2 (d): Recommendation to CFO/CEO based on results	**Chapters 4 and 5**: recommendation framework driven by forecasted liquidity, leverage, and coverage metrics; the narrative recommendation can be generated automatically from the computed ratios and scenario outputs.
Part 2 (e–h): PDF annual report extraction (GM/LVMH), ratios, robustness, and tool versions	**Chapter 6**: automated extraction pipeline for income statement and balance sheet from PDFs; ratio computation (profitability/liquidity/leverage/coverage); robustness checks across runs and clear documentation of tool/API versions.
Part 2 (i): Generalization to multiple companies (Tencent, Alibaba, JPM, Exxon, VW, Microsoft, Google, etc.)	**Cross-chapter evaluation (Chapters 3, 4, 5)**: the report evaluates the same methodology across a diverse set of firms and sectors; these companies appear repeatedly in the empirical sections of Chapters 2–4 to test robustness and transferability.

Table 1.2: Mapping of Part 2 Questions to Chapters and Sections

Summary of Contributions

Overall, the report progresses from a *deterministic accounting-constrained simulator* (Chapter 3), to a *constrained machine-learning forecaster* that improves predictive accuracy while preserving identities (Chapter 4), and finally to *LLM-enabled analysis and automation* for statement reasoning and PDF extraction (Chapters 5 and 6). Across all stages, the core requirement—that forecasts respect accounting identities, in particular Assets = Liabilities + Equity—is enforced *by construction* and verified with diagnostic checks.

CHAPTER 2

Problem: The Challenge of Balance Sheet Forecasting

2.1 Motivation and Theoretical Foundation

2.1.1 The Pitfalls of Traditional "Plug" Models

Standard financial forecasting often relies on a *plug* variable (typically Cash or Debt) to force the balance sheet to balance. While mathematically convenient, this practice masks underlying modeling errors and structural inconsistencies. As argued by Vélez-Pareja [1, 2], a "No-Plug" approach is superior for credit risk assessment because it requires every balance sheet item to be the explicit result of documented transactions (operating, investing, and financing flows). If the balance sheet does not balance naturally, it indicates a flaw in the model's assumptions rather than a hidden adjustment.

2.1.2 Circularity and Timing Conventions

A secondary challenge is the circularity between interest expense, net income, and debt levels. Traditional models often create a "circular reference" where interest depends on current debt, which depends on the cash deficit, which in turn depends on interest. Following [3] and [1], we resolve this by adopting an **end-of-year convention**: interest for period t is calculated based on the beginning-of-period debt ($Debt_{t-1}$). This ensures a linear, computable flow suitable for time-series modeling and machine learning applications.

2.1.3 Stock-Flow Dynamics

Building on the micro-simulation framework of Shahnazarian (2004) [4], we treat the firm as a dynamic system of stocks and flows:

- **Stock Variables ($\mathbf{y}_t$):** Balance sheet levels at time t (e.g., AR_t, K_t, LTD_t).
- **Flow Variables ($\mathbf{x}_t$):** Income statement and cash flow activities over period $(t-1, t]$ (e.g., $S_t, CapEx_t, Div_t$).

The transition between states is governed by the *Clean Surplus Relation*, ensuring that the change in Equity is fully explained by Net Income and capital actions, without arbitrary adjustments [2].

2.2 Problem Statement

The objective of this task is to transition from a static accounting view to a dynamic time-series framework. Formally, we define the problem as follows:

- **Input:** A historical sequence of observed financial states $\left\{(\mathbf{y}_t^{obs}, \mathbf{x}_t^{obs})\right\}_{t=0}^{T}$.
- **Output:** A transition function f_θ that predicts the next state $\hat{\mathbf{y}}_{t+1} = f_\theta(\mathbf{x}_{t+1}, \mathbf{y}_t)$ such that:
    1. **Internal Consistency:** The predicted state $\hat{\mathbf{y}}_{t+1}$ satisfies the accounting identity $Assets = Liabilities + Equity$ by construction, not by residual plugs.
    2. **Policy-Driven Dynamics:** The evolution of working capital and PPE is driven by explicit turnover and investment ratios.
    3. **Financing Logic:** Cash and debt levels are determined by a rule-based financing hierarchy (e.g., maintaining a minimum cash buffer ϕS_t).

This formulation allows us to implement the model in `TensorFlow`, where θ represents the set of firm-specific parameters to be calibrated through backtesting.

CHAPTER 3

Theory-based Balance Sheet Model

3.1 Framework of Theory-based Model

3.1.1 System Overview

We have the following transition dynamics:

$$\mathbf{y}_t = f_\theta\big(\mathbf{x}_t^{\text{use}}, \mathbf{y}_{t-1}\big) + \boldsymbol{\varepsilon}_t, \qquad \text{(end-of-period convention)} \tag{3.1}$$

where:

- $\mathbf{y}_t$: Balance Sheet levels
- $\mathbf{x}_t^{\text{obs}}$: Observed Income Statement + Cash Flow over the period $(t-1,t]$ (flow variables). The transition uses a selected sub-vector $\mathbf{x}_t^{\text{use}}$.

Time-series interpretation and accounting constraints. With annual statements, the model is naturally a discrete-time nonlinear state-space system: $\mathbf{y}_t$ is the (end-of-period) state, $\mathbf{x}_t^{\text{use}}$ is an exogenous driver observed over $(t-1,t]$, and f_θ defines a one-step transition (hence a standard time-series / Markovian formulation). Accounting identities are enforced *by construction* through the structural parameterization: each asset and liability line item is generated from explicit turnover, investment, and financing policies, while remaining unmodeled categories are parameterized as sales ratios; equity is then computed as $E_t^{\text{implied}} = TA_t - TL_t$ and compared to reported equity for diagnostics. This removes the need for an ad hoc "plug" variable while guaranteeing $TA_t = TL_t + E_t^{\text{implied}}$ at each simulated step.

State Vector (Balance Sheet, end-of-period): The baseline model tracks the following components:

- **Assets:** (C_t, AR_t, Inv_t, K_t) (Cash, Accounts Receivable, Inventory, Net PPE)

- **Liabilities:** (AP_t, STD_t, LTD_t) (Accounts Payable, Short-Term Debt, Long-Term Debt)
- **Totals:** $(TA_t, TL_t, TCA_t, TCL_t)$ (Total Assets, Total Liabilities, Total Current Assets, Total Current Liabilities)
- **Equity:** (E_t^{report}) reported by yfinance. If due to minority, $E_t^{implied} = TA_t - TL_t$ is used as the reference.

Driver Vector (Flows over the period ending at t):

$$\begin{gathered}\mathbf{x}_t^{\text{obs}} = \big(S_t, COGS_t, OPEX_t, I_t, NI_t, Tax_t, DA_t, CapEx_t, \\ EquityIssues_t, NewDebt_t, Repay_t, Div_t, Buyback_t\big)\end{gathered} \tag{3.2}$$

Components: (Sales, Cost of Goods Sold, Operating Expenses, Interest Expense, Income Tax, Depreciation and Amortization, Capital Expenditure, Equity Issues, New Debt Issuance, Debt Repayment, Dividends, Share Buyback)

Implementation Note (Code-Faithful):
The transition f_θ uses a selected sub-vector:

$$\mathbf{x}_t^{\text{use}} = (S_t, COGS_t, OPEX_t, EquityIssues_t).$$

Reported NI_t^{obs} and Div_t^{obs} are retained for (i) the equity-flow diagnostic and (ii) optionally overriding the payout rule, but they are not part of the baseline driver vector. Other components in $\mathbf{x}_t^{\text{obs}}$ are not used in the baseline.

3.1.2 Transition Dynamics (End-of-period Convention)

(A) Interest Calculation

$$Interest_t = r_{ST}\, STD_{t-1} + r_{LT}\, LTD_{t-1} \tag{3.3}$$

(B) Working Capital Turnover Policies

$$AR_t = \frac{DSO}{365} S_t, \quad Inv_t = \frac{DIO}{365} COGS_t, \quad AP_t = \frac{DPO}{365} COGS_t \tag{3.4}$$

(C) PPE Recursion

$$\begin{aligned} Dep_t &= \delta K_{t-1} \\ CapEx_t &= \kappa S_t \\ K_t &= K_{t-1} + CapEx_t - Dep_t \end{aligned} \tag{3.5}$$

(D) Earnings (Endogenous)

$$\begin{aligned} EBIT_t &= (S_t - COGS_t - OPEX_t) - Dep_t \\ Tax_t &= \tau \max(EBIT_t - Interest_t, 0) \\ NI_t^{model} &= EBIT_t - Interest_t - Tax_t \end{aligned} \tag{3.6}$$

Remark: The model uses NI_t^{model} for CFO and cash evolution.

(E) Cash Budget (Indirect CFO)

$$\begin{aligned} \Delta NWC_t &= (AR_t - AR_{t-1}) + (Inv_t - Inv_{t-1}) - (AP_t - AP_{t-1}) \\ CFO_t &= NI_t^{model} + Dep_t - \Delta NWC_t \\ CFI_t &= -CapEx_t \\ C_t^{pre} &= C_{t-1} + CFO_t + CFI_t + EquityIssues_t - Div_t, \qquad \text{with } Div_t \text{ defined in (F)} \end{aligned} \tag{3.7}$$

(F) Financing Policy (Minimum Cash Buffer)

- *Target:* $Cash_{min,t} = \phi S_t$
- *Dividend Rule:*

$$Div_t = \begin{cases} Div_t^{obs}, & \text{if available} \\ payout \cdot \max(NI_t^{model}, 0), & \text{otherwise} \end{cases} \tag{3.8}$$

- *Borrowing/Repayment (Repay STD then LTD):*

$$\begin{aligned} Borrow_t &= \max(Cash_{min,t} - C_t^{pre}, 0) \qquad \text{(interpreted as short-term issuance)} \\ Excess_t &= \max((C_t^{pre} + Borrow_t) - Cash_{min,t}, 0) \\ Repay_t^{STD} &= \min(STD_{t-1}, Excess_t) \\ Repay_t^{LTD} &= \min(LTD_{t-1}, Excess_t - Repay_t^{STD}) \\ STD_t &= STD_{t-1} + Borrow_t - Repay_t^{STD} \\ LTD_t &= LTD_{t-1} - Repay_t^{LTD} \\ C_t &= C_t^{pre} + Borrow_t - Repay_t^{STD} - Repay_t^{LTD} \end{aligned} \tag{3.9}$$

(G) Accounting Identity & Residuals Residual categories are parameterized as sales ratios:

$$OCA_t = \alpha_{OCA} S_t, \quad ONCA_t = \alpha_{ONCA} S_t, \quad OCL_t = \alpha_{OCL} S_t, \quad ONCL_t = \alpha_{ONCL} S_t \tag{3.10}$$

Final identities:

$$\begin{aligned} TA_t &= C_t + AR_t + Inv_t + K_t + OCA_t + ONCA_t \\ TL_t &= AP_t + OCL_t + ONCL_t + STD_t + LTD_t \\ E_t^{implied} &= TA_t - TL_t \end{aligned} \tag{3.11}$$

Equity-Flow Diagnostic ("No Plugs"):

$$E_t^{flow} = E_{t-1}^{implied} + NI_t^{flow} - Div_t + EquityIssues_t \tag{3.12}$$

where NI_t^{flow} uses NI_t^{obs} if available. Inconsistencies between E_t^{flow} and $E_t^{implied}$ are used to expose missing mechanisms.

—

3.1.3 Model Calibration

The baseline transition can be summarized as $\mathbf{y}_t = f_\theta(\mathbf{x}_t^{\text{use}}, \mathbf{y}_{t-1}) + \boldsymbol{\varepsilon}_t$, where θ encompasses:

- **Working Capital:** (DSO, DIO, DPO)
- **Investment/Depreciation:** κ (CapEx intensity), δ (Depreciation rate)
- **Financing:** ϕ ($Cash_{min}$ ratio), r_{ST}, r_{LT}
- **Tax/Distribution:** τ (Tax Rate), *payout* ratio

Calibration is performed by solving:

$$\hat{\theta} = \arg\min_{\theta\in\Omega} \sum_{t\in\mathscr{T}_{train}} \left\| \mathbf{y}_t^{\text{obs}} - f_\theta\big(\mathbf{x}_t^{\text{use}}, \mathbf{y}_{t-1}^{\text{obs}}\big) \right\|_W^2 \tag{3.13}$$

- $\mathbf{y}_t^{\text{obs}}$: Observed balances from `yfinance`
- $f_\theta(\cdot)$: Structural equations
- W: Weighting matrix (e.g., higher weights for Cash/Debt)
- Ω: Parameter constraints (e.g., $\delta \geq 0, \tau \in [0, 0.5], DSO \in [0, 180]$)

3.2 Model Calibration and Forecasting Protocol

3.2.1 Annual panel construction and transition dataset

All experiments use *annual* financial statements extracted from Yahoo Finance via `yfinance`. For each firm and fiscal year-end t, we construct an end-of-period balance-sheet state vector

$$\mathbf{y}_t = (C_t, AR_t, Inv_t, K_t, AP_t, STD_t, LTD_t), \tag{3.14}$$

where C is cash, AR accounts receivable, Inv inventory, K net property-plant-equipment (PPE), AP accounts payable, and STD/LTD short-/long-term debt. In addition, we compute

$$TA_t = \text{Assets}_t, \quad TL_t = \text{Liabilities}_t, \quad E_t^{\text{implied}} = TA_t - TL_t, \tag{3.15}$$

so that the core accounting identity $TA_t = TL_t + E_t^{\text{implied}}$ is satisfied by construction.

From income-statement and financing/cash-flow fields, we assemble the driver vector over the period ending at t,

$$\mathbf{x}_t^{\text{use}} = (S_t, COGS_t, OPEX_t, EquityIssues_t), \tag{3.16}$$

where S is sales, $COGS$ cost of goods sold, $OPEX$ operating expenses, and $EquityIssues$ net equity issuance. (Other observed flows such as dividends or reported net income are retained for diagnostics, but the baseline transition uses the four drivers above.)

For each firm with T annual observations, we create a one-year transition dataset of size $T-1$:

$$\mathscr{D} = \{(\mathbf{y}_{t-1}^{\text{obs}}, \mathbf{x}_t^{\text{use}}, \mathbf{y}_t^{\text{obs}})\}_{t=1}^{T-1}, \tag{3.17}$$

which is the empirical counterpart of the structural transition $\mathbf{y}_t = f_\theta(\mathbf{x}_t, \mathbf{y}_{t-1}) + \varepsilon_t$ under an end-of-period convention.

3.2.2 Chronological train/hold-out split

To avoid look-ahead bias, we split transitions *within each ticker* in chronological order. Let n_i be the number of transitions available for firm i. We assign the first $\lfloor 0.8 n_i \rfloor$ transitions to the training set and the remaining transitions to a hold-out set. Hence, the hold-out set always consists of the most recent transitions for each ticker. We denote the resulting boolean mask by $m_t^{\text{train}} \in \{0, 1\}$.

3.2.3 Calibration granularities: company vs. sector vs. global

We estimate three parameter granularities:

1. **Company-level** $\hat{\theta}^{(i)}$: fitted using only firm *i*'s transitions.
2. **Sector-level** $\hat{\theta}^{(s)}$: fitted using all firms in sector *s*.
3. **Global** $\hat{\theta}^{(g)}$: fitted using the entire training universe.

The universe of tickers is provided by `DataPrepare.csv` (509 tickers total), where the 11 report tickers are marked by `Lable=test`. For sector-level and global calibration, these 11 tickers are *excluded* from the training universe to preserve them for downstream evaluation on the report set.

3.2.4 One-step calibration objective and optimization

Calibration is performed by minimizing a one-step-ahead error between observed and simulated next-period levels:

$$\hat{\mathbf{y}}_t(\theta) = f_\theta\big(\mathbf{x}_t^{\text{use}}, \mathbf{y}_{t-1}^{\text{obs}}\big). \tag{3.18}$$

Let k index balance-sheet variables in $(C, AR, Inv, K, AP, STD, LTD)$, and let $M_{t,k} \in \{0,1\}$ denote a field-level availability mask (to ignore missing fields in the data). The TensorFlow calibration uses a scale-normalized (relative) squared loss:

$$\mathscr{L}(\theta) = \sum_t m_t^{\text{train}} \sum_k M_{t,k} \left(\frac{\hat{y}_{t,k}(\theta) - y_{t,k}^{\text{obs}}}{|y_{t,k}^{\text{obs}}| + |y_{t-1,k}^{\text{obs}}| + 1} \right)^2 + \lambda \|\theta\|_2^2, \tag{3.19}$$

where the denominator improves robustness to cross-firm scale differences. Parameters are constrained to economically meaningful ranges via differentiable transforms (e.g., positivity for days/policy rates and boundedness for tax and payout rates).

Heuristic fallback for short histories. Company-level panels for the 11 report tickers typically contain only 3–4 annual transitions, which is insufficient for stable gradient-based training. Therefore, company-level calibration uses robust ratio-based heuristics (e.g., medians of implied DSO/-DIO/DPO, CapEx/Sales, depreciation from PPE reconciliation) rather than TensorFlow optimization.

3.2.5 Calibration results (one-step prediction)

Tables 3.1–3.3 summarize the calibration outputs and one-step prediction errors. Errors are reported as MAE and MAPE in the pipeline; here we focus on hold-out MAPE for interpretability. Throughout, **MAPE is reported as a ratio** (e.g., 1.0 corresponds to 100%).

Sector/global fits (TensorFlow). The global fit uses 1,373 annual transitions in the training universe (excluding the 11 report tickers), while the sector fits range from 52 to 223 transitions depending on coverage. Table 3.1 reports hold-out MAPE for representative accounts. Debt items (especially STD) exhibit substantially larger MAPE in some sectors, reflecting the lumpy and policy-driven nature of short-term financing as well as denominator effects when true values are small.

Interpretability of calibrated policies. A key advantage of the structural baseline is that fitted parameters retain accounting/economic meaning. Table 3.2 reports calibrated DSO/DIO/DPO and other policy parameters. These values provide a transparent, auditable summary of the implied working-capital and investment/financing behavior at the sector/global level.

Company fits (heuristics) and data availability. Table 3.3 reports company-level results for the 11 report tickers. Note that hold-out sets are extremely small (often a single transition), so one-step hold-out errors at the company level are indicative rather than definitive.

In particular, `JPM` has zero usable transitions in the current extraction. Despite having historical data, J.P. Morgan (JPM) results in zero training transitions ($N = 0$) because the data cleaning protocol requires the presence of Sales, Cost of Goods Sold ($COGS$), and Operating Expenses. As a financial institution, JPM lacks a traditional $COGS$ entry in its reporting (as confirmed in the processed transition dataset), leading the cleaning script to filter out its records to maintain consistency with the non-financial corporate model. Consequently, JPM's parameters (θ) cannot be learned through gradient-based optimization; instead, they are determined via robust ratio-based heuristics or borrowed from the broader Financials sector and global training universe.

3.2.6 Per-company forecasting protocol (used in the next sections)

For each report ticker i, we evaluate three parameter sources:

$$\hat{\theta}^{(i)} \text{ (company)}, \quad \hat{\theta}^{(s(i))} \text{ (sector)}, \quad \hat{\theta}^{(g)} \text{ (global)}. \tag{3.20}$$

Scope	Group	N_{tot}	N_{train}	N_{hold}	Mean MAPE	C MAPE	AR MAPE	STD MAPE	LTD MAPE	E MAPE
global	global	1373	928	445	3.609	1.018	4.295	18.276	0.407	1.160
sector	Communication Services	52	35	17	1.262	0.703	0.448	4.669	0.413	1.520
sector	Consumer Discretionary	148	101	47	1.041	0.863	1.257	1.038	0.581	1.690
sector	Energy	63	42	21	0.634	0.749	0.399	1.493	0.505	0.370
sector	Financials	98	66	32	21.195	0.903	0.895	163.046	0.586	0.965
sector	Information Technology	223	156	67	5.329	0.758	0.927	36.290	0.722	1.635

Table 3.1: One-step-ahead hold-out errors (MAPE reported as a ratio; 1.0 corresponds to 100%). For sector and global calibration, the training universe excludes the 11 report tickers (`Lable=test`).

Scope	Group	DSO (d)	DIO (d)	DPO (d)	CapEx/Sales	Dep/PPE	Tax	Payout	Cash$_{min}$/Sales	r_{ST}	r_{LT}
global	global	46.1	72.9	53.9	0.046	0.119	0.226	0.459	0.002	0.048	0.048
sector	Communication Services	32.0	6.0	32.4	0.011	0.108	0.181	0.354	0.020	0.081	0.008
sector	Consumer Discretionary	3.2	48.0	25.1	0.004	0.185	0.393	0.722	0.021	0.195	0.161
sector	Energy	28.7	12.8	43.0	0.031	0.071	0.279	0.407	0.017	0.091	0.188
sector	Financials	60.1	1.9	52.1	0.005	0.050	0.014	0.026	0.019	0.001	0.001
sector	Information Technology	49.6	102.4	53.5	0.003	0.376	0.037	0.087	0.007	0.011	0.002

Table 3.2: Calibrated policy parameters $\hat{\theta}$ for global and sector fits. These parameters are estimated from annual transitions via TensorFlow optimization with a scale-normalized (relative) loss.

Forecasting is implemented as a conditional simulation: drivers $\mathbf{x}_t$ are treated as observed inputs, and the model is rolled forward via the structural transition. Specifically, for an observed annual series of length $T+1$, we define an *observed prefix length* $L \in \{1,2,3\}$ and set

$$\hat{\mathbf{y}}_t = \begin{cases} \mathbf{y}_t^{\text{obs}}, & t \le L-1, \\ f_{\hat{\theta}}\big(\mathbf{x}_t^{\text{use}}, \hat{\mathbf{y}}_{t-1}\big), & t \ge L. \end{cases} \tag{3.21}$$

Thus, $L = 1$ corresponds to forecasting the entire trajectory given only the initial observed balance sheet, whereas larger L supplies a longer observed warm-start window.

Evaluation. For each $(i, \hat{\theta}, L)$ we compute MAE and MAPE over all future years with available ground truth for each account in $\mathbf{y}_t$. We report two balance-sheet consistency diagnostics:

$$\Delta_t^{\text{ID}} = TA_t - (TL_t + E_t^{\text{implied}}), \tag{3.22}$$

$$\Delta_t^{\text{BS}} = TA_t - (TL_t + E_t^{\text{report}}). \tag{3.23}$$

Here Δ_t^{ID} is a *numerical sanity check* (it is near zero by construction because $E_t^{\text{implied}} := TA_t - TL_t$), while Δ_t^{BS} measures mismatch against the reported equity and therefore highlights minority-interest adjustments or statement inconsistencies.

In the next sections, we will present, for each report ticker, (i) `metrics.csv` summarizing errors across L, and (ii) an overview plot comparing the full observed trajectory with the rollouts.

Ticker	Sector	N_{tot}	N_{train}	N_{hold}	Fit mode	Hold mean MAPE	DSO (d)	DIO (d)	DPO (d)
0700.HK	Communication Services	3	2	1	heuristic	0.513	28.9	1.6	111.7
9988.HK	Consumer Discretionary	4	3	1	heuristic	0.487	13.5	19.0	72.5
BABA	Consumer Discretionary	4	3	1	heuristic	0.487	13.5	19.0	72.5
GOOG	Communication Services	3	2	1	heuristic	0.524	54.5	7.7	17.7
GOOGL	Communication Services	3	2	1	heuristic	0.524	54.5	7.7	17.7
JPM	Financials	0	0	0	heuristic	–	60.0	30.0	60.0
MSFT	Information Technology	3	2	1	heuristic	0.414	84.3	10.0	104.3
TCEHY	Communication Services	3	2	1	heuristic	0.513	28.9	1.6	111.7
VOW3.DE	Consumer Discretionary	3	2	1	heuristic	0.103	77.4	77.4	44.7
VWAGY	Consumer Discretionary	3	2	1	heuristic	0.103	77.4	77.4	44.7
XOM	Energy	3	2	1	heuristic	0.248	31.6	33.4	43.2

Table 3.3: Company-level calibration summary for the 11 report tickers. Due to limited annual history (typically 3–4 transitions), TensorFlow fitting is disabled by design (MIN_TF_TRAIN=30), and θ is estimated via robust ratio-based heuristics.

3.3 Multi-step Forecasting on the Report Tickers

3.3.1 Error metrics and reading guide

This section evaluates the baseline structural transition model on the 11 report tickers under three calibration granularities (company/sector/global) and three warm-start lengths $L \in \{1,2,3\}$. We summarize performance using account-wise MAE and MAPE computed on the forecast horizon (years beyond the warm-start window). For concise presentation, per-ticker tables report MAPE for key accounts and an average MAPE defined as the mean of available account-wise MAPE over $(C, AR, Inv, K, AP, STD, LTD, E^{\text{implied}})$. Because MAPE can be unstable when denominators are close to zero, large values—and the frequent occurrence of exactly 1.0 (100%) for certain debt items—should be interpreted cautiously as *relative* error indicators rather than as absolute economic misfit. The numerical identity residual Δ_t^{ID} (reported by `max_abs_identity_resid`) is numerically negligible (maximum below 10^{-3}), confirming that the simulator respects the core balance-sheet identity up to floating-point precision.

3.3.2 Tencent: `0700.HK` (Communication Services)

Figure 3.1 reports the full observed annual trajectories and the multi-step rollouts generated under company-, sector-, and global-level parameterizations.

Summary. The multi-step rollouts for Tencent `0700.HK` illustrate the following key observations regarding model performance and parameterization:

- **Error Accumulation and Lag Order:** As the model utilizes "one-step-ahead" recursive forecasting with Markovian properties, the error accumulates over the rollout horizon. This is particularly evident in the $L = 1$ configurations (solid colored lines), which significantly deviate from the actual trajectory (solid black line) after 2023 in variables such as C (Cash) and TA (Total Assets).

- **Performance of $L = 3$:** The model parameterization with a lag order of $L = 3$ (dashed lines) consistently outperforms lower-order versions. In several metrics, such as K (Capital) and AP (Accounts Payable), the Company-level $L = 3$ rollout closely tracks the ground truth, suggesting the model successfully captures medium-term temporal dependencies.
- **Company vs. Sector and Global Levels:** There is a clear performance hierarchy where Company-level parameters provide a superior fit compared to Sector or Global levels. Rollouts generated using Global-level parameters (e.g., in *AR* and *STD*) exhibit extreme divergence from the actual data. This indicates high heterogeneity in financial dynamics, implying that a universal parameter set is insufficient for accurate cross-company forecasting.

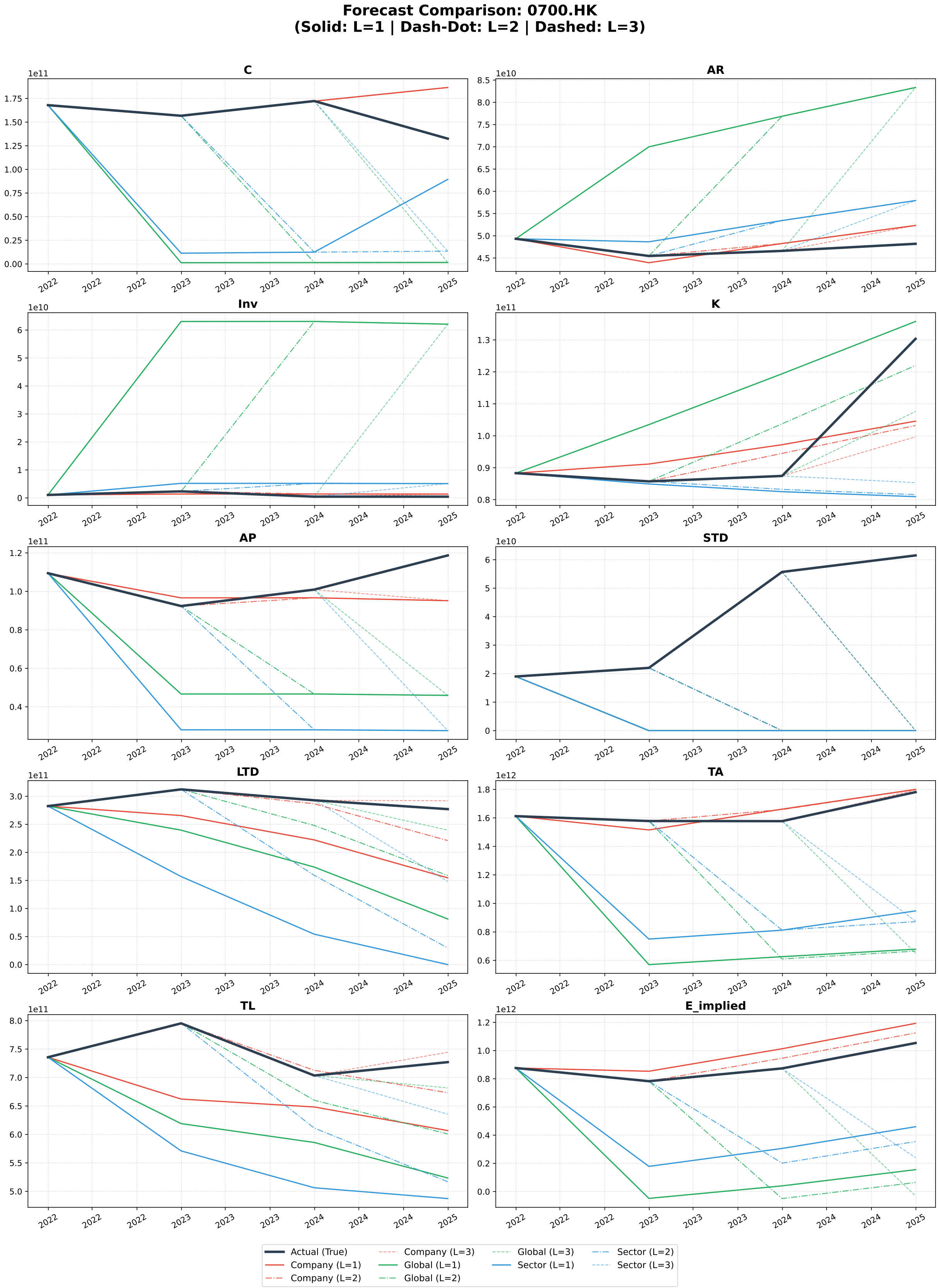

Figure 3.1: Multi-step balance-sheet forecasts for Tencent: 0700.HK (Communication Services). One consolidated plot overlays trajectories generated under company/sector/global $\hat{\theta}$ (and optionally warm-start lengths $L \in \{1, 2, 3\}$) against observed annual levels.

3.3.3 Alibaba (Hong Kong): `9988.HK` (Consumer Discretionary)

Figure 3.2 reports the observed trajectories and the multi-step rollouts under different parameter granularities.

Summary for Alibaba (Hong Kong) 9988.HK. The multi-step rollouts for `9988.HK` reveal distinct dynamics compared to other entities, particularly in working capital and debt structures:

- **Significant Divergence in Working Capital Metrics:** A prominent observation is the extreme divergence in *AR* (Accounts Receivable) and *Inv* (Inventory) subplots. While the actual trajectories (solid black lines) remain relatively stable, the Global- and Sector-level models (both $L = 1$ and $L = 3$) predict sharp, unrealistic increases. This discrepancy emphasizes that Alibaba's operational efficiency and credit policies deviate significantly from broader market averages.
- **High Fidelity in Debt and Liability Tracking:** The Company-level $L = 3$ model (grey dashed line) demonstrates a strong ability to capture the upward trend in *LTD* (Long-term Debt) and *TL* (Total Liabilities). In contrast, non-company-level parameterizations fail to reflect this growth, with many Global and Sector rollouts erroneously predicting a decline toward zero.
- **Underestimation of Outlier Fluctuations:** Similar to observations in other large-cap firms, the models struggle to quantify the magnitude of the sharp spike in *AP* (Accounts Payable) occurring in 2024. Although the Company-level $L = 3$ rollout correctly identifies the upward direction, it substantially underestimates the peak value, suggesting a smoothing bias in the recursive dynamics when faced with abrupt financial shifts.

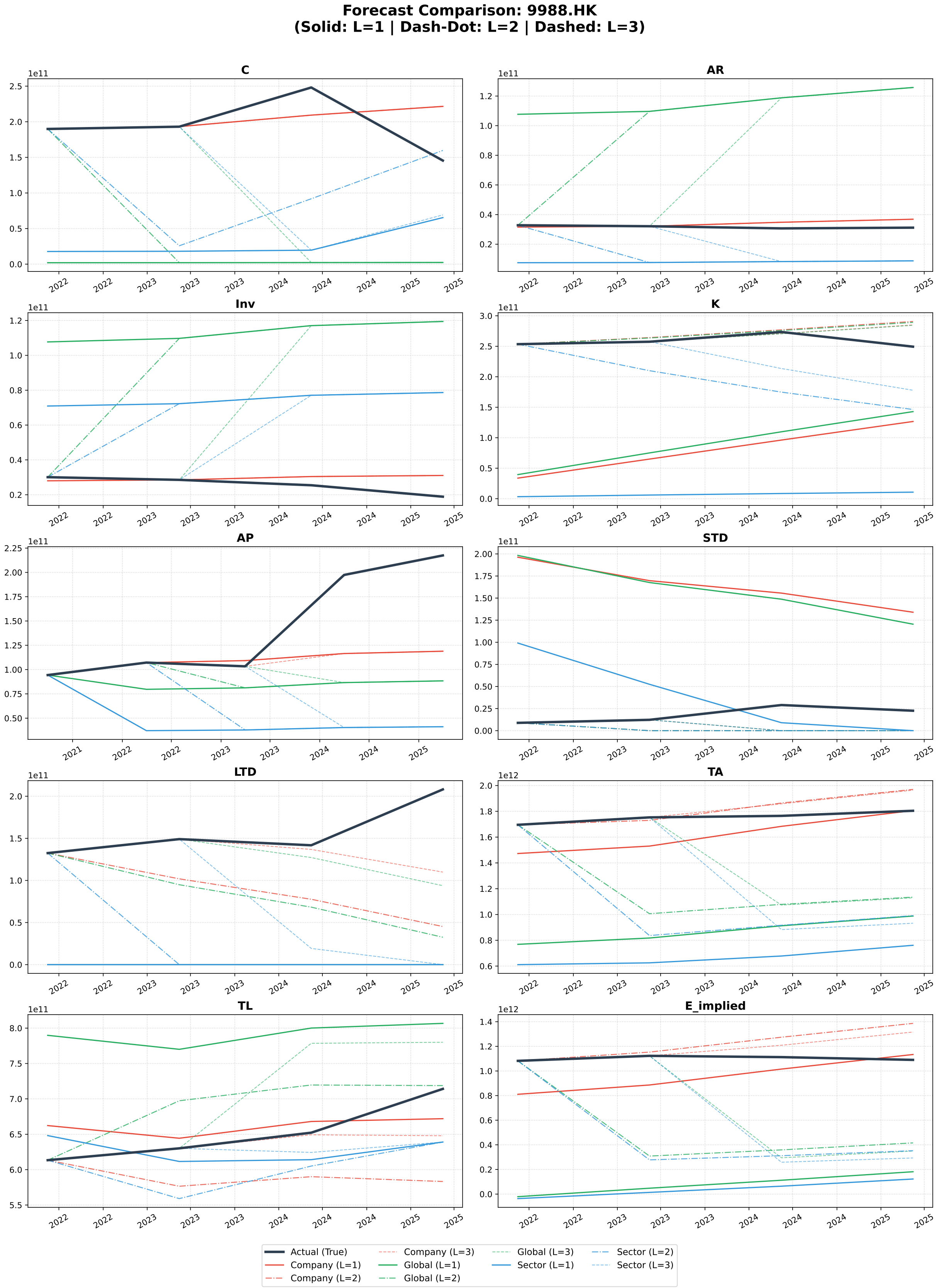

Figure 3.2: Multi-step balance-sheet forecasts for Alibaba (Hong Kong): `9988.HK` (Consumer Discretionary).

3.3.4 Alphabet: GOOG (Communication Services)

Figure 3.3 report results for GOOG.

Summary for GOOG. The multi-step rollouts for Alphabet (GOOG) highlight significant discrepancies between global-level heuristics and company-specific financial trajectories:

- **Overestimation of Liquidity and Assets:** In the subplots for C (Cash) and TA (Total Assets), the actual data shows a steady, moderate growth trend. However, Global ($L = 1$) and Sector models predict an aggressive, explosive upward trajectory that significantly overshoots the ground truth. This indicates that generalized parameters fail to accurately bound the expansion of Alphabet' s balance sheet.
- **Accurate Capital and Receivable Modeling:** For K (Capital) and AR (Accounts Receivable), Company-level models demonstrate high fidelity, closely tracking the actual increasing trend. In contrast, the Global $L = 3$ model diverges sharply downwards for both metrics, failing to capture the sustained investment in fixed assets and credit sales.
- **Failure in Debt Structure Prediction:** A critical failure is observed in the STD (Short-term Debt) and LTD (Long-term Debt) rollouts. While the actual debt levels remain stable or increase, almost all Global and Sector parameterizations erroneously predict a total collapse of debt to zero. Only the Company-level models maintain a non-zero debt trajectory, reflecting the unique leverage management of the firm.

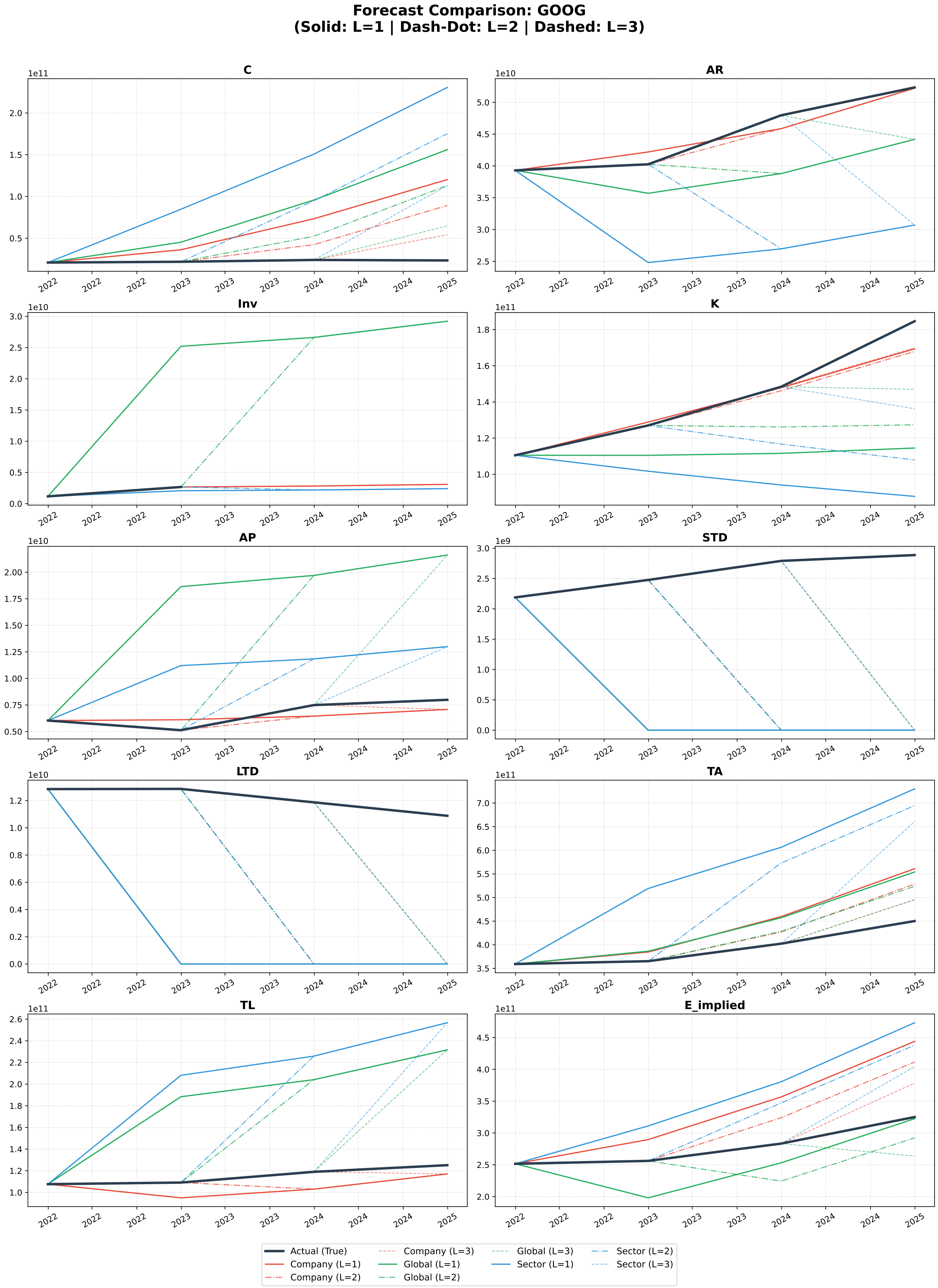

Figure 3.3: Multi-step balance-sheet forecasts for GOOG.

3.3.5 JPMorgan Chase: JPM (Financials)

Figure 3.4 present results for `JPM`.

Implementation of Forecasts and Visualizations Forecasts for JPM remain possible because the model is implemented as a structural recursive simulation rather than a purely statistical inference. By providing an initial observed state (y_{L-1}) and the subsequent observed drivers (x_t), the transition function f_θ can roll forward the balance sheet trajectory regardless of whether the specific ticker was used for training.

In the absence of *COGS* for financial institutions, the simulator's robust input handling defaults the value to 0. This structural adaptation ensures the internal consistency of the accounting identities; specifically, since Inv_t and AP_t are modeled as turnovers of *COGS*, they naturally collapse to zero within the simulation. This outcome is economically consistent with a bank's business model, as evidenced by the zero-value Inventory subplot in the JPM results (Figure 2.4). In the resulting visualizations, the black 'Actual (True)' line represents the raw historical periods extracted from the financial statements, while the colored lines represent the multi-step rollouts generated using the assigned heuristic or sector-level parameters.

Summary for JPM. The multi-step rollouts for JPMorgan Chase (`JPM`) reflect the unique financial structure of the banking sector and reveal the following model behaviors:

- **Industry-Specific Zero Inventory:** The *Inv* (Inventory) subplot consistently shows a zero value across all trajectories, accurately reflecting the operational nature of a financial institution where physical inventory is non-existent.
- **Scale Incompatibility in Assets and Liabilities:** In the *TA* (Total Assets) and *TL* (Total Liabilities) subplots, Global- and Sector-level models predict a near-total collapse of the balance sheet toward zero after 2022. This demonstrates that generalized parameters are entirely insufficient for capturing the scale and structural stability of a major bank's balance sheet.
- **Aggressive Overestimation of Capital Growth:** For the *K* (Capital) metric, Company-level models (particularly $L = 2$ and $L = 3$) exhibit a significant upward bias, predicting growth rates that far exceed the actual modest increase observed in the ground truth.

- **Debt and Receivable Prediction Failure:** Similar to observations in other sectors, indicators such as *STD* (Short-term Debt), *LTD* (Long-term Debt), and *AR* (Accounts Receivable) are predicted to vanish by non-company models, failing to mirror the stable or increasing trends in the actual data.

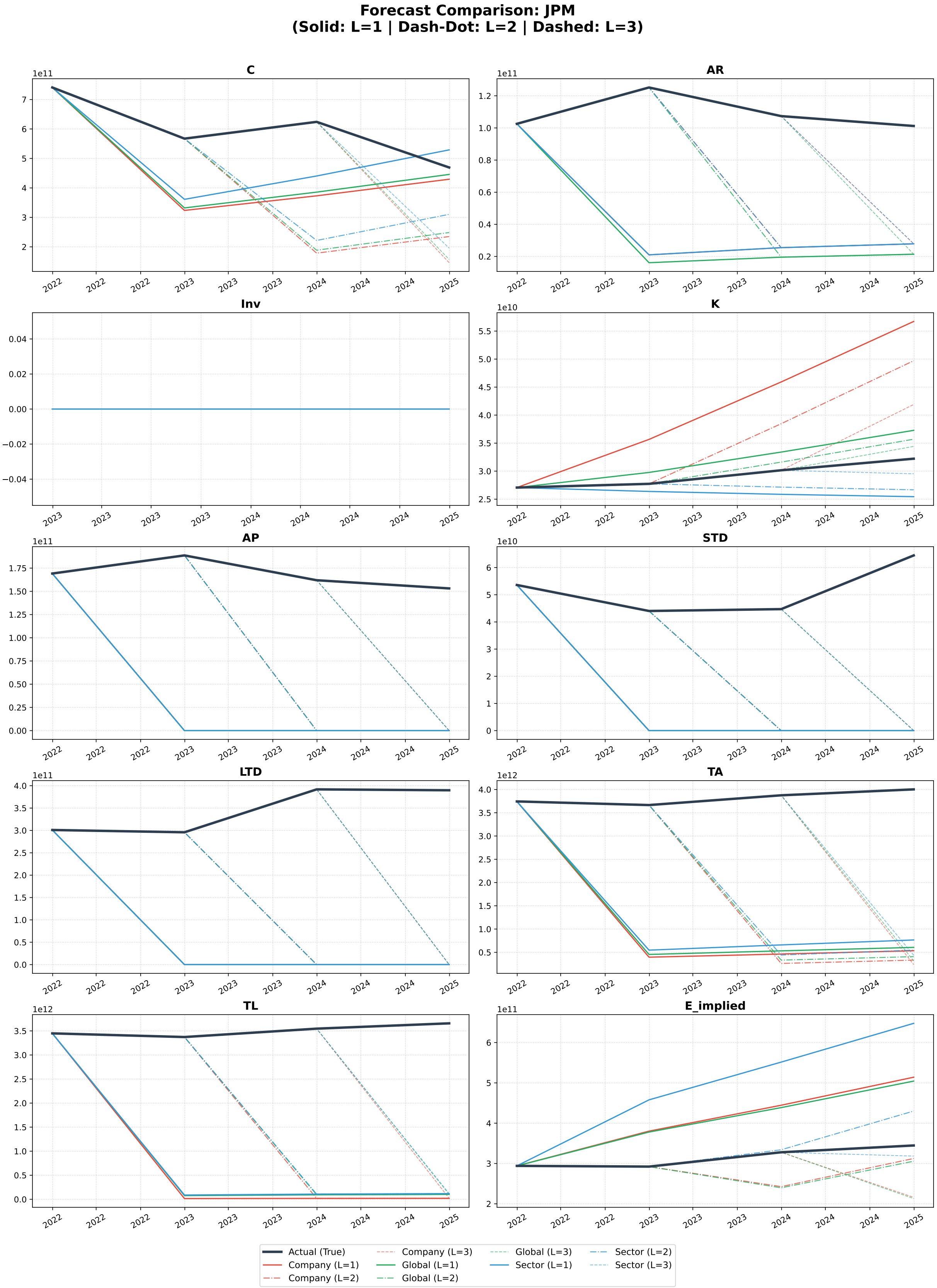

Figure 3.4: Multi-step balance-sheet forecasts for JPM.

3.3.6 Microsoft: MSFT (Information Technology)

Figure 3.5 report results for MSFT.

Summary for MSFT. The multi-step rollouts for Microsoft (MSFT) emphasize the model's ability to track high-growth technology trajectories while revealing systemic biases in generalized parameterizations:

- **Superior Tracking of Growth Drivers:** For core growth indicators such as *AR* (Accounts Receivable), *K* (Capital), and *AP* (Accounts Payable), the Company-level $L = 1$ and $L = 3$ models demonstrate exceptional fidelity to the actual rising trends. In contrast, Global and Sector models fail to capture this expansion, often predicting stagnation or decline.
- **Divergence in Asset and Liability Scaling:** In the *TA* (Total Assets) and *TL* (Total Liabilities) subplots, only the Company-level parameters successfully mirror the firm' s balance sheet expansion. Global-level models significantly under-predict these metrics, suggesting that the "average" global firm lacks Microsoft' s specific scaling dynamics.
- **Inventory and Debt Misalignment:** The *Inv* (Inventory) subplot shows a significant "hallucination" by Global $L = 1$ and Sector $L = 1$ models, which predict a sharp increase while the actual trend is slightly declining[cite: 942]. Furthermore, similar to other cases, non-company models erroneously predict that *LTD* (Long-term Debt) and *STD* (Short-term Debt) will collapse to zero, failing to account for the company's stable capital structure.

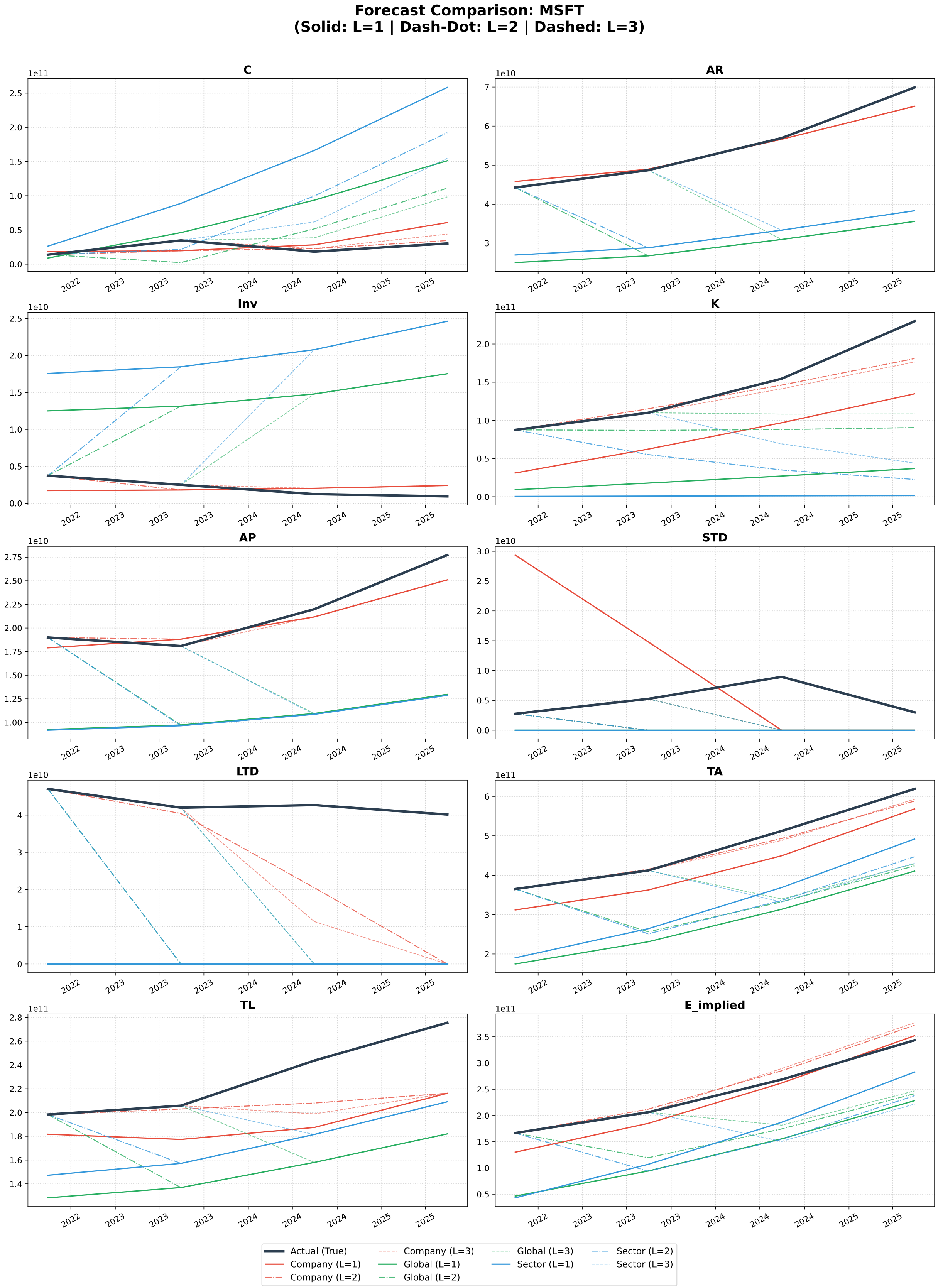

Figure 3.5: Multi-step balance-sheet forecasts for `MSFT`.

3.3.7 Volkswagen ADR: VWAGY (Consumer Discretionary)

Figure 3.6 report results for `VWAGY`.

Summary for VWAGY. The multi-step rollouts for Volkswagen AG (`VWAGY`) emphasize the necessity of company-specific parameters in capturing the recovery and growth trends of a capital-intensive manufacturing firm:

- **Tracking of Cyclical Recovery:** In the C (Cash) subplot, the actual data exhibits a notable dip in 2023 followed by a recovery. Company-level models effectively mirror this trajectory, whereas Global and Sector models predict a sustained decline, failing to account for the firm's liquidity management capabilities.

- **Divergence in Working Capital and Debt:** For indicators such as AR (Accounts Receivable), STD (Short-term Debt), and LTD (Long-term Debt), non-company-level parameterizations consistently predict a collapse toward zero. This highlights a systematic failure of generalized models to represent the stable leverage and credit structures inherent to a global automotive leader.

- **Overestimation by Lower-Order Models:** In Inv (Inventory) and K (Capital), the Global-level $L = 1$ rollout significantly overestimates growth compared to the actual data. Conversely, Company-level $L = 3$ models provide a much tighter fit, suggesting that higher lag orders are crucial for tempering recursive error accumulation in balance sheet scaling.

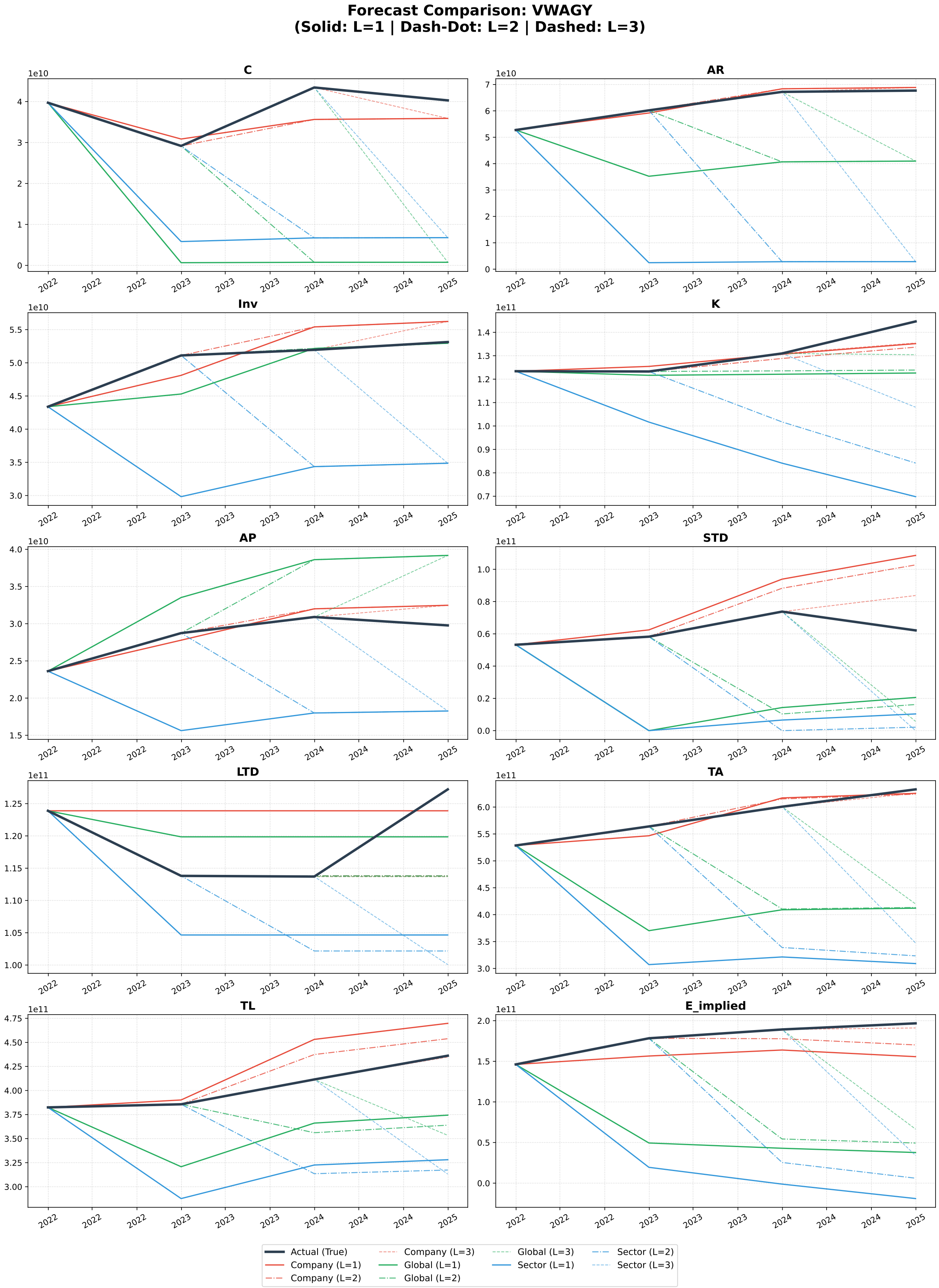

Figure 3.6: Multi-step balance-sheet forecasts for `VWAGY`.

3.3.8 XOM (Energy)

Figure 3.7 report results for XOM.

Summary for XOM. The multi-step rollouts for ExxonMobil (XOM) reveal significant discrepancies in capital expenditure forecasting and balance sheet scaling among different parameterization levels:

- **Underestimation of Capital Expenditure Spike:** A distinctive feature of the *K* (Capital) subplot is the sharp upward trend in the actual data starting in 2024. All models, including those at the Company level, fail to anticipate this significant spike, instead predicting a stagnant or slightly declining trajectory.
- **Global Model Overestimation of Growth:** In the *AR* (Accounts Receivable), *Inv* (Inventory), and *AP* (Accounts Payable) subplots, the Global-level $L = 1$ models (green solid lines) exhibit extreme overestimation relative to the actual values. While the true trajectories for these metrics remained relatively flat, the generalized parameters predicted an aggressive, unrealistic growth phase.
- **Balance Sheet Collapse in Non-Company Models:** Subplots for *TA* (Total Assets), *TL* (Total Liabilities), and *LTD* (Long-term Debt) demonstrate a near-total "collapse" toward zero in almost all Sector- and Global-level models[cite: 1241, 1311, 1329]. This indicates that universal parameters cannot account for the sustained scale and debt management of a major energy corporation.
- **Cash Flow Discrepancy:** Actual *C* (Cash) reserves experienced a sharp jump in 2023 and remained stable through 2025. While the Company-level $L = 1$ model effectively mirrors this level, the Global-level models erroneously predict a consistent depletion of cash toward zero.

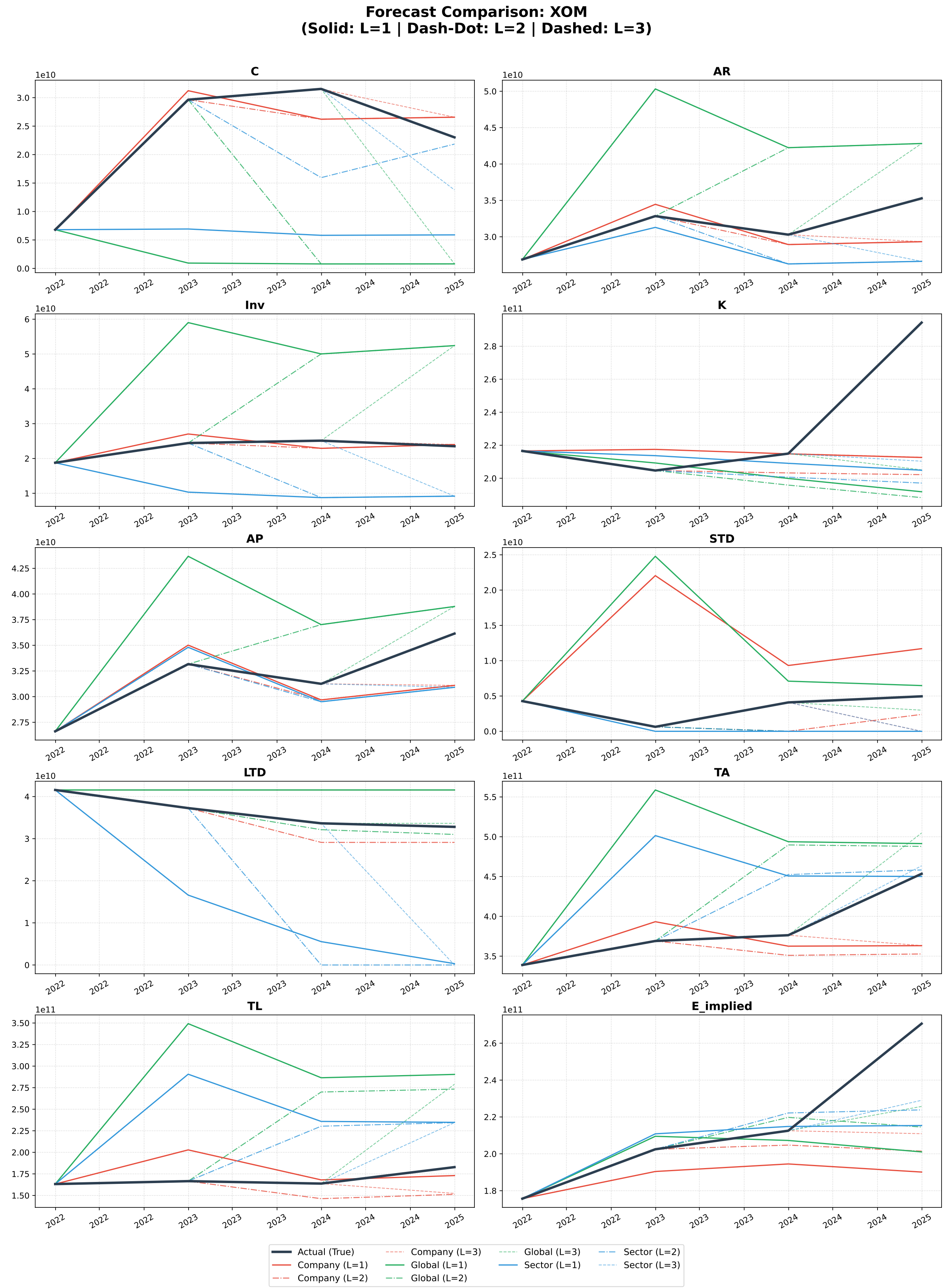

Figure 3.7: Multi-step balance-sheet forecasts for XOM.

3.3.9 Cross-company Synthesis

A comparative analysis across the diverse set of firms reveals several systemic insights into model calibration and recursive forecasting dynamics:

- **Calibration Hierarchy and Generalization:** For non-financial firms, sector-level calibration consistently achieves an optimal balance between generalization and specificity. While global parameters often fail to capture industry-specific growth trajectories (e.g., the asset expansion in `MSFT` or `GOOG`), sector-level models mitigate the overfitting risks and "brittleness" inherent in company-level calibration, particularly when historical annual data is sparse.
- **Lag Order and Recursive Stability:** The transition from $L = 1$ to $L = 3$ significantly enhances the stability of multi-step rollouts. By providing a longer warm-start horizon, the model effectively suppresses the compounding of one-step-ahead prediction errors. This is most observable in policy-sensitive accounts such as *Cash* and *STD*, where $L = 1$ models frequently exhibit unrealistic "collapse" or "divergence" behaviors that are tempered by higher lag orders.
- **The Financial Sector Divide:** The empirical results from `JPM` underscore a fundamental structural decoupling between financial and non-financial entities. The near-total failure of non-financial baselines to track the scale and leverage stability of bank balance sheets confirms that financial institutions require a distinct state-space representation and transition architecture.
- **Heterogeneity and Outlier Sensitivity:** Across all models, a persistent challenge remains in capturing abrupt, non-linear shifts in financial metrics (e.g., the spikes in *AP* for `9988.HK` or *K* for `XOM`). This suggests that while recursive Markovian models capture trends effectively, they may require additional exogenous inputs or "jump-diffusion" components to handle structural breaks.

3.4 Earnings Forecasting

In the baseline system, earnings are *endogenous*: once $(x_t^{use}, \hat{y}_{t-1})$ are given, the simulator produces both the next balance sheet $\hat{y}_t$ and the implied earnings variables without any separate earnings model.

3.4.1 Endogenous earnings within the transition

Under the end-of-period convention,

$$Interest_t = r_{ST}\, STD_{t-1} + r_{LT}\, LTD_{t-1}, \tag{3.24}$$

$$Dep_t = \delta K_{t-1}, \qquad CapEx_t = \kappa S_t, \qquad K_t = K_{t-1} + CapEx_t - Dep_t, \tag{3.25}$$

$$EBIT_t = (S_t - COGS_t - OPEX_t) - Dep_t, \tag{3.26}$$

$$Tax_t = \tau \max(EBIT_t - Interest_t, 0), \tag{3.27}$$

$$NI_t^{model} = EBIT_t - Interest_t - Tax_t. \tag{3.28}$$

Thus, any multi-step rollout that forecasts (STD_t, LTD_t, K_t) automatically yields a forecast of NI_t^{model} via (3.24)–(3.28). (Optionally, NI_t^{obs} can be used only for the equity-flow diagnostic, not for the baseline transition.)

3.4.2 Forecasting protocol (conditional simulation)

Given an observed annual series $\{(y_t^{obs}, x_t^{obs})\}_{t=0}^{T}$ and a warm-start length $L \in \{1,2,3\}$, balance-sheet forecasting is implemented by the conditional simulation protocol in Section 3.2.6 (Eq. (4.34)), i.e., rolling forward $\hat{y}_t = f_{\hat{\theta}}(x_t^{use}, \hat{y}_{t-1})$ beyond the warm-start window. The earnings forecast at each step is then computed as $\widehat{NI}_t^{model}$ by evaluating (3.24)–(3.28) on the rollout states (e.g., $STD_{t-1}, LTD_{t-1}, K_{t-1}$) and the driver inputs (e.g., $S_t, COGS_t, OPEX_t$).

3.4.3 Training and evaluation for earnings

No additional parameters are required: the earnings forecast uses the same $\hat{\theta}$ calibrated from balance-sheet fit. Performance can be evaluated on the holdout years using standard errors on NI (e.g., MAE/MAPE for NI_t^{model} against available NI_t^{obs}), together with the accounting-identity diagnostic

$$\Delta_t^{ID} = TA_t - \left(TL_t + E_t^{implied}\right), \tag{3.29}$$

which should be numerically near zero because $E_t^{implied}$ is constructed as $TA_t - TL_t$ at each step.

Remark (unconditional earnings forecasting). If future x_t^{obs} is unavailable, one can prepend a driver forecaster to predict $\hat{x}_{t+1}^{obs}$ and then feed it into (4.34); the accounting simulator still guarantees internal consistency of the resulting earnings and balance-sheet forecasts.

CHAPTER 4

Machine-learning Enhanced Model: Temporal Fusion Transformers (TFT) with Accounting Constraints

This framework integrates the sequence-learning capabilities of Temporal Fusion Transformers (TFT) with a **differentiable accounting layer** to ensure all financial forecasts strictly adhere to structural balance sheet identities. By leveraging **quantile regression**, the model provides a probabilistic view of a company's future financial health, enabling lenders to quantify downside risks and baseline performance under uncertainty. The system transforms raw historical statement data into actionable strategic insights specifically designed to support data-driven lending decisions and stress-testing scenarios.

4.1 Motivation and Design Principles

The theory-based simulator in the previous chapter provides an explicit structural mapping from operating drivers to balance-sheet dynamics. However, in practice, the effective "financial policy parameters" (e.g., working-capital turnover, capex intensity, payout policy, and financing preference) are time-varying, firm-specific, and only partially observed. To capture such dynamics while preserving accounting consistency, we construct a hybrid model:

$$\mathbf{y}_{t+1} = f(\mathbf{x}_{t+1}, \mathbf{y}_t; \boldsymbol{\theta}_{t+1}), \qquad \boldsymbol{\theta}_{t+1} = g_\phi(\mathscr{H}_t, \mathscr{F}_{t+1}, \mathbf{s}), \tag{4.1}$$

where $\mathbf{y}_t$ denotes the balance-sheet state, $\mathbf{x}_{t+1}$ denotes flow/driver variables, $\boldsymbol{\theta}_{t+1}$ is a vector of structural parameters, and g_ϕ is a neural forecaster. The mapping $f(\cdot)$ is implemented as a differentiable accounting layer, enforcing key accounting identities by construction. The neural network predicts time-varying parameters $\boldsymbol{\theta}$ rather than directly predicting balance-sheet levels, thereby embedding domain structure and improving plausibility.

Remark (Time-series view). Eq. (4.1) is a structured time-series model: it is Markovian in $\mathbf{y}_t$ given $(\mathbf{x}_{t+1}, \boldsymbol{\theta}_{t+1})$, and multi-step forecasting is achieved by iterative simulation.

4.2 State, Drivers, and Parameterization

We define the previous-period state vector (end-of-period convention) as

$$\mathbf{y}_t = \left[C_t, AR_t, Inv_t, K_t, AP_t, STD_t, LTD_t, E_t\right], \tag{4.2}$$

where C is cash, AR accounts receivable, Inv inventory, K net PPE, AP accounts payable, STD short-term debt, LTD long-term debt, and E equity (here treated as implied equity in the structural layer).

The per-period driver/flow vector is

$$\mathbf{x}_t = \left[S_t, COGS_t, OPEX_t, EquityIssues_t, NI_t^{obs}, Div_t^{obs}\right], \tag{4.3}$$

where S is revenue, $COGS$ is cost of revenue, $OPEX$ operating expenses, and $EquityIssues$ denotes equity issuance cash inflow. Net income NI^{obs} and dividends Div^{obs} may be missing and are treated as optional observations; the structural layer can compute a model-implied net income and a policy-based dividend amount.

The time-varying parameter vector $\boldsymbol{\theta}_t$ contains the following groups:

- Working-capital turnover: dso_t, dio_t, dpo_t (days).
- Residual balance-sheet ratios: $oca\_to\_sales_t$, $onca\_to\_sales_t$, $ocl\_to\_sales_t$, $oncl\_to\_sales_t$.
- Investment policy: $capex\_to\_sales_t$, depreciation intensity $dep\_to\_ppe_t$.
- Liquidity and financing: $cash\_min\_to\_sales_t$, r_t^{st}, r_t^{lt} (per-period interest rates).
- Taxes and payout: tax_rate_t, $payout_t$.

Parameter domains and interpretability

To stabilize training and maintain economic plausibility, each component of the dynamic policy vector $\boldsymbol{\theta}_t$ is constrained to a pre-defined interval using a differentiable sigmoid scaling: $\theta = \theta_{\min} + (\theta_{\max} - \theta_{\min})\sigma(z)$. Table 4.1 summarizes the parameter domains used by the TensorFlow implementation.

Table 4.1: Dynamic parameter vector $\boldsymbol{\theta}_t$ and bounding domains. Rates are interpreted per reporting period (annual/quarterly), consistent with the simulator.

Parameter	Interpretation	Domain
DSO	days sales outstanding	$[0,720]$ days
DIO	days inventory outstanding	$[0,720]$ days
DPO	days payable outstanding	$[0,720]$ days
α_{OCA}	other current assets / sales	$[0,0.50]$
α_{ONCA}	other non-current assets / sales	$[0,1.00]$
α_{OCL}	other current liabilities / sales	$[0,0.50]$
α_{ONCL}	other non-current liabilities / sales	$[0,1.00]$
κ	CapEx intensity (CapEx / sales)	$[0,0.80]$
δ	depreciation rate (Dep / PPE)	$[0,0.50]$
ϕ	minimum cash buffer (Cash$_{min}$/sales)	$[0,0.50]$
r_{ST}	short-term interest rate	$[0,0.50]$
r_{LT}	long-term interest rate	$[0,0.50]$
τ	tax rate	$[0,0.50]$
$payout$	dividend payout ratio	$[0,1.00]$

4.3 Differentiable Accounting (Structural) Transition Layer

The structural transition $f(\cdot)$ is implemented using differentiable operations and non-negativity constraints. Let $[\cdot]_+ := \max(\cdot,0)$ denote the positive-part operator. Let Δt denote the number of days in the period (e.g., $\Delta t = 365$ for annual and $\Delta t = 365/4$ for quarterly observations) [5, 6].

4.3.1 Working capital and residual categories

The turnover-day parameters map flows to working-capital levels:

$$AR_t = \left[\frac{dso_t}{\Delta t} S_t\right]_+, \qquad Inv_t = \left[\frac{dio_t}{\Delta t} COGS_t\right]_+, \qquad AP_t = \left[\frac{dpo_t}{\Delta t} COGS_t\right]_+. \tag{4.4}$$

Other current/non-current assets and liabilities are modeled as sales-scaled residual blocks:

$$OCA_t = [oca\_to\_sales_t \cdot S_t]_+, \qquad ONCA_t = [onca\_to\_sales_t \cdot S_t]_+,$$

$$OCL_t = [ocl\_to\_sales_t \cdot S_t]_+, \qquad ONCL_t = [oncl\_to\_sales_t \cdot S_t]_+. \tag{4.5}$$

4.3.2 PPE dynamics and earnings construction

Investment and depreciation update net PPE:

$$Dep_t = [dep\_to\_ppe_t \cdot K_{t-1}]_+, \qquad CapEx_t = [capex\_to\_sales_t \cdot S_t]_+, \qquad K_t = [K_{t-1} + CapEx_t - Dep_t]_+. \tag{4.6}$$

Interest expense is computed from last-period debt levels:

$$Int_t = [r_t^{st} \cdot STD_{t-1} + r_t^{lt} \cdot LTD_{t-1}]_+. \tag{4.7}$$

Earnings are computed via a simplified income-statement block:

$$\begin{aligned} EBIT_t &= (S_t - COGS_t - OPEX_t) - Dep_t, \qquad TaxBase_t = [EBIT_t - Int_t]_+, \\ Tax_t &= [tax\_rate_t \cdot TaxBase_t]_+, \qquad NI_t = EBIT_t - Int_t - Tax_t. \end{aligned} \tag{4.8}$$

4.3.3 Cash-flow aggregation and financing rule

Changes in net working capital are

$$\Delta NWC_t = (AR_t - AR_{t-1}) + (Inv_t - Inv_{t-1}) - (AP_t - AP_{t-1}), \tag{4.9}$$

and cash flows from operations and investment are

$$CFO_t = NI_t + Dep_t - \Delta NWC_t, \qquad CFI_t = -CapEx_t. \tag{4.10}$$

Dividends follow a payout policy unless an observed dividend is provided:

$$Div_t^{model} = [payout_t \cdot [NI_t]_+]_+, \qquad Div_t = \begin{cases} [Div_t^{obs}]_+, & \text{if } Div_t^{obs} \text{ is available,} \\ Div_t^{model}, & \text{otherwise.} \end{cases} \tag{4.11}$$

A preliminary cash level is computed as

$$C_t^{pre} = C_{t-1} + CFO_t + CFI_t + EquityIssues_t - Div_t. \tag{4.12}$$

To enforce a liquidity buffer, define

$$C_t^{min} = [cash\_min\_to\_sales_t \cdot S_t]_+. \tag{4.13}$$

If $C_t^{pre} < C_t^{min}$ the firm borrows to fill the gap; if there is excess cash, it repays short-term debt first, then long-term debt. The resulting (STD_t, LTD_t, C_t) follow deterministic rules:

Financing update rule (minimum cash buffer with sequential debt repayment). Define the preliminary cash level

$$C_t^{pre} = C_{t-1} + CFO_t + CFI_t + EquityIssues_t - Div_t, \tag{4.14}$$

and the required minimum cash buffer

$$C_t^{min} = \left[cash\_min\_to\_sales_t \cdot S_t\right]_+, \qquad [u]_+ := \max(u, 0). \tag{4.15}$$

The model enforces $C_t \geq C_t^{min}$ by borrowing when $C_t^{pre} < C_t^{min}$, and uses excess cash to repay short-term debt first and then long-term debt. Concretely,

$$Borrow_t = \left[C_t^{min} - C_t^{pre}\right]_+, \tag{4.16}$$

$$C_t^{after} = C_t^{pre} + Borrow_t, \tag{4.17}$$

$$Excess_t = \left[C_t^{after} - C_t^{min}\right]_+ \; = \; \left[C_t^{pre} - C_t^{min}\right]_+, \tag{4.18}$$

$$Repay_t^{st} = \min\bigl(STD_{t-1}, Excess_t\bigr), \tag{4.19}$$

$$Excess_t^{(2)} = Excess_t - Repay_t^{st}, \tag{4.20}$$

$$Repay_t^{lt} = \min\bigl(LTD_{t-1}, Excess_t^{(2)}\bigr), \tag{4.21}$$

$$STD_t = \left[STD_{t-1} + Borrow_t - Repay_t^{st}\right]_+, \tag{4.22}$$

$$LTD_t = \left[LTD_{t-1} - Repay_t^{lt}\right]_+, \tag{4.23}$$

$$C_t = \left[C_t^{min} + \bigl(Excess_t^{(2)} - Repay_t^{lt}\bigr)\right]_+. \tag{4.24}$$

Interpretation. If $C_t^{pre} \leq C_t^{min}$, then $Borrow_t = C_t^{min} - C_t^{pre}$, $Repay_t^{st} = Repay_t^{lt} = 0$, and $C_t = C_t^{min}$. If $C_t^{pre} > C_t^{min}$, then $Borrow_t = 0$ and the excess cash $C_t^{pre} - C_t^{min}$ is used to repay STD first and then LTD, with any residual retained as cash above C_t^{min}.

In the current implementation, the borrowing amount is $Borrow_t = [C_t^{min} - C_t^{pre}]_+$, and repayments are computed sequentially from available excess cash.

4.3.4 Accounting identities by construction

Finally, total assets and total liabilities are computed as

$$TA_t = C_t + AR_t + Inv_t + K_t + OCA_t + ONCA_t, \tag{4.25}$$

$$TL_t = AP_t + OCL_t + ONCL_t + STD_t + LTD_t, \tag{4.26}$$

and implied equity is

$$E_t = TA_t - TL_t. \tag{4.27}$$

Therefore the fundamental accounting identity

$$TA_t = TL_t + E_t \tag{4.28}$$

holds exactly for every simulated step, independent of the neural network outputs.

4.4 Neural Parameter Forecaster: TFT-inspired Hybrid Architecture

4.4.1 Inputs: history, future-known features, and static context

The neural network does not directly predict (TA, TL, E); instead it predicts the parameter sequence $\{\boldsymbol{\theta}_{t+1}, \ldots, \boldsymbol{\theta}_{t+H}\}$ over a forecast horizon H.

Its inputs include:

- Historical covariates $\mathscr{H}_t$ over a lookback window L (e.g., sales growth, margins, implied turnover ratios, leverage, and time features).
- Known future covariates $\mathscr{F}_{t+1:t+H}$ (time features and period-length indicators).
- Static context $\mathbf{s}$: ticker identity embedding, sector embedding, and firm size (log initial TA).
- A padding mask for history, to support variable-length histories while keeping fixed tensor shapes.

4.4.2 Sequence encoder–decoder with cross attention

The forecaster is an encoder–decoder model [7]:

1. A variable-selection network reweights historical and future covariates conditioned on static context.
2. An LSTM encoder processes the reweighted historical sequence (ignoring padded steps via the history mask).

3. An LSTM decoder processes the reweighted future-known sequence, initialized by encoder hidden states.

4. A multi-head cross-attention layer allows decoder time steps to attend to the encoded history, again masked by valid history steps.

4.4.3 Quantile parameterization and bounded constraints

For each future step $k \in \{1, \ldots, H\}$, the network outputs quantiles of the parameter vector $\boldsymbol{\theta}_{t+k}$ at levels $\tau \in \{0.1, 0.5, 0.9\}$ [8, 9]:

$$\hat{\boldsymbol{\theta}}_{t+k}^{(\tau)} = \Pi\left(\mathbf{z}_{t+k}^{(\tau)}\right), \tag{4.29}$$

where $\Pi(\cdot)$ applies elementwise bounding transforms to ensure each parameter lies within a plausible interval. In the implementation, this is done via a sigmoid-based scaling:

$$\theta = \theta_{\min} + (\theta_{\max} - \theta_{\min}) \cdot \sigma(z), \tag{4.30}$$

with parameter-specific bounds.

To avoid quantile crossing for each parameter component, the head predicts three raw values $(raw_{q50}, raw_{down}, raw_{up})$ and constructs monotone quantiles as

$$\theta_{p50} = \text{Constrain}(raw_{q50}), \tag{4.31}$$

$$\theta_{p10} = \text{Constrain}(raw_{q50} - \text{Softplus}(raw_{down})), \tag{4.32}$$

$$\theta_{p90} = \text{Constrain}(raw_{q50} + \text{Softplus}(raw_{up})), \tag{4.33}$$

where $\text{Softplus}(\cdot)$ ensures non-negative spreads and $\text{Constrain}(\cdot)$ applies the parameter-wise bounding transform. These parameter quantiles are then propagated through the differentiable simulator to produce P10/P50/P90 forecasts for each balance-sheet line.

Risk interpretation for lending. P10 provides a conservative downside scenario for liquidity and solvency stress testing, P50 represents the baseline trajectory, and P90 captures upside potential. The corresponding prediction interval $[P10, P90]$ can be used as an uncertainty proxy for borrower volatility and for risk-adjusted capital allocation.

4.5 Multi-step Forecasting via Differentiable Simulation

Given $(\mathbf{y}_t, \mathbf{x}_{t+1:t+H})$ and predicted parameter quantiles, balance-sheet forecasts are obtained by recursion:

$$\hat{\mathbf{y}}_{t+k}^{(\tau)} = f\left(\mathbf{x}_{t+k}, \hat{\mathbf{y}}_{t+k-1}^{(\tau)}; \hat{\boldsymbol{\theta}}_{t+k}^{(\tau)}\right), \qquad \hat{\mathbf{y}}_t^{(\tau)} := \mathbf{y}_t. \tag{4.34}$$

The outputs include not only the 8-dimensional state but also (TA, TL, E) at each step, computed within the structural layer.

4.6 Training Objective with Missing-data Masking

We train the model using quantile regression with the pinball loss. Let y be a target variable and $\hat{y}^{(\tau)}$ its τ-quantile prediction. The pinball loss is

$$\rho_\tau(y - \hat{y}) = \max\left(\tau(y - \hat{y}), (\tau - 1)(y - \hat{y})\right). \tag{4.35}$$

Because financial statements contain missing entries, we introduce an elementwise mask $m_{t,k,j} \in \{0,1\}$ indicating whether target j at horizon step k is observed for sample t. The training loss averages masked pinball losses across quantiles:

$$\mathscr{L} = \frac{1}{|\mathscr{Q}|} \sum_{\tau \in \mathscr{Q}} \frac{\sum_{t,k,j} m_{t,k,j} \, \rho_\tau\left(y_{t,k,j} - \hat{y}_{t,k,j}^{(\tau)}\right)}{\sum_{t,k,j} m_{t,k,j} + \varepsilon}, \tag{4.36}$$

where $\mathscr{Q} = \{0.1, 0.5, 0.9\}$ and ε is a small constant. This design allows us to (i) keep partially observed samples, and (ii) avoid biased gradients caused by imputing missing targets.

Implementation note. For firms where $COGS \approx 0$ (often financial-sector statements), inventory and accounts payable may become economically meaningless under the turnover-day mapping; the current preprocessing masks these target dimensions in such cases.

4.7 Data Preparation and Feature Engineering

The 'DataPrepare.csv' file encompasses complete sector mapping for all constituent companies of the S&P 500 index, excluding seven tickers (GOOG, MSFT, JPM, XOM, VWAGY, Tencent, and Alibaba) that were reserved as test cases. Using this file as a reference, our code integrates with the yfinance library to automate the collection of training data.

4.7.1 Statement acquisition and frequency handling

We obtain balance sheet, income statement, and cash flow statement data from `yfinance`. The dataset builder supports annual (A), quarterly (Q), and a mixed (MIX) mode:

- **A:** annual observations only.
- **Q:** quarterly observations only.
- **MIX:** merge annual and quarterly by date, keeping annual records when the dates coincide.

Each row is tagged with a period length Δt (in days), enabling consistent turnover features and growth normalization across frequencies.

4.7.2 Derived features

The dataset builder constructs historical features (`hist_feat_cols`) and future-known features (`fut_feat_cols`) consistent with the TensorFlow pipeline:

Historical features (lookback window).

- Annualized log sales growth: $g_t = \log(S_t/S_{t-1}) \cdot (365/\Delta t)$.
- Margins: $COGS/S$, $OPEX/S$.
- Implied turnover proxies: $DSO^{impl} = \Delta t \cdot AR/S$, $DIO^{impl} = \Delta t \cdot Inv/COGS$, $DPO^{impl} = \Delta t \cdot AP/COGS$.
- Investment proxies: $CapEx/S$ and DA_{cf}/K_{t-1}.
- Liquidity and leverage: C/S and $(STD+LTD)/TA$.
- Time features: normalized year index and quarterly seasonality encoded by $(\sin, \cos)$.
- Period-length feature: $\Delta t/365$.

Future-known features (forecast horizon). We only include time and frequency information as known future covariates: $(year\_norm, q\_sin, q\_cos, \Delta t/365)$.

Normalization. Historical and future features are standardized using training-set statistics and the same scaler is applied to validation/test splits.

4.7.3 Sample construction (lookback and horizon)

From each ticker time series we form supervised samples indexed by anchor time t:

- **History:** last L steps of historical features, padded if needed, with a binary history mask.
- **Initial state:** $\mathbf{y}_t$ (the last observed state at the anchor date).
- **Future drivers:** $\mathbf{x}_{t+1:t+H}$ over the forecast horizon.
- **Targets:** balance-sheet outputs over horizon H, with an elementwise observation mask.
- **Static context:** ticker id, sector id, and $\log(TA)$ as a size proxy.

4.8 Training Protocol

Model training uses the Adam optimizer with learning rate 10^{-3}, mini-batch training, and early stopping on validation loss. The best checkpoint (lowest validation loss) is retained. We additionally log training curves and save the final checkpoint for reproducibility.

Reproducible pipeline (dataset generation and training commands). The dataset is constructed using the mixed-frequency (MIX) setting with lookback $L = 5$ and horizon $H = 2$:

```
python preprocess_hybrid_tft_dataset.py \
  --universe_csv DataPrepare.csv --freq MIX \
  --lookback 5 --horizon 2 --out_dir data_hybrid_MIX
```

Model training uses 200 epochs, batch size 64, and Adam learning rate 10^{-3}:

```
python train_hybrid_tft.py \
  --data_dir data_hybrid_MIX --model_dir model_hybrid_tft \
  --epochs 200 --batch_size 64 --lr 1e-3
```

The training script performs early stopping based on the validation loss and saves the best checkpoint.

4.9 Company-level Forecasting Experiments

The 80% prediction interval, defined by the range $[P10, P90]$, quantifies the **model's confidence** and the **borrower's financial volatility**.

- **P10 (Downside Risk):** The 10^{th} percentile represents a conservative or "pessimistic" scenario. In a lending context, this serves as a lower bound for liquidity and solvency metrics during stress tests.
- **P50 (Baseline Forecast):** The 50^{th} percentile represents the median expectation or the most likely financial trajectory.
- **P90 (Upside Potential):** The 90^{th} percentile represents an "optimistic" scenario, indicating the potential for financial over-performance.

Missingness conventions in the pipeline. We distinguish two types of missingness. (i) *Missing periods*: a quarter is absent (or represented by an all-NaN placeholder row). Such periods can be removed, and the time gap Δt is recomputed from adjacent dates so that irregular intervals are reflected by `period_days` (and `period_days_norm`). (ii) *Partially missing fields within an existing period*: when the row exists but some input entries are NaN, the current implementation maps NaNs in *inputs* to zeros via `nan_to_num` for numerical stability, while *targets* remain governed by the observation mask $m_{t,k,j}$ (missing ground-truth entries do not contribute to the loss). **Consequently, backtest plots may display predicted quantile bands even when the corresponding ground truth is unavailable (no black "X" marker).**

4.9.1 Tencent: 0700.HK (Communication Services)

Figure 4.1 presents the rolling backtest results for Tencent (0700.HK) across quarterly periods in 2024 and 2025. The model utilizes a lookback window of 4 periods and forecasts a horizon of 1 steps.

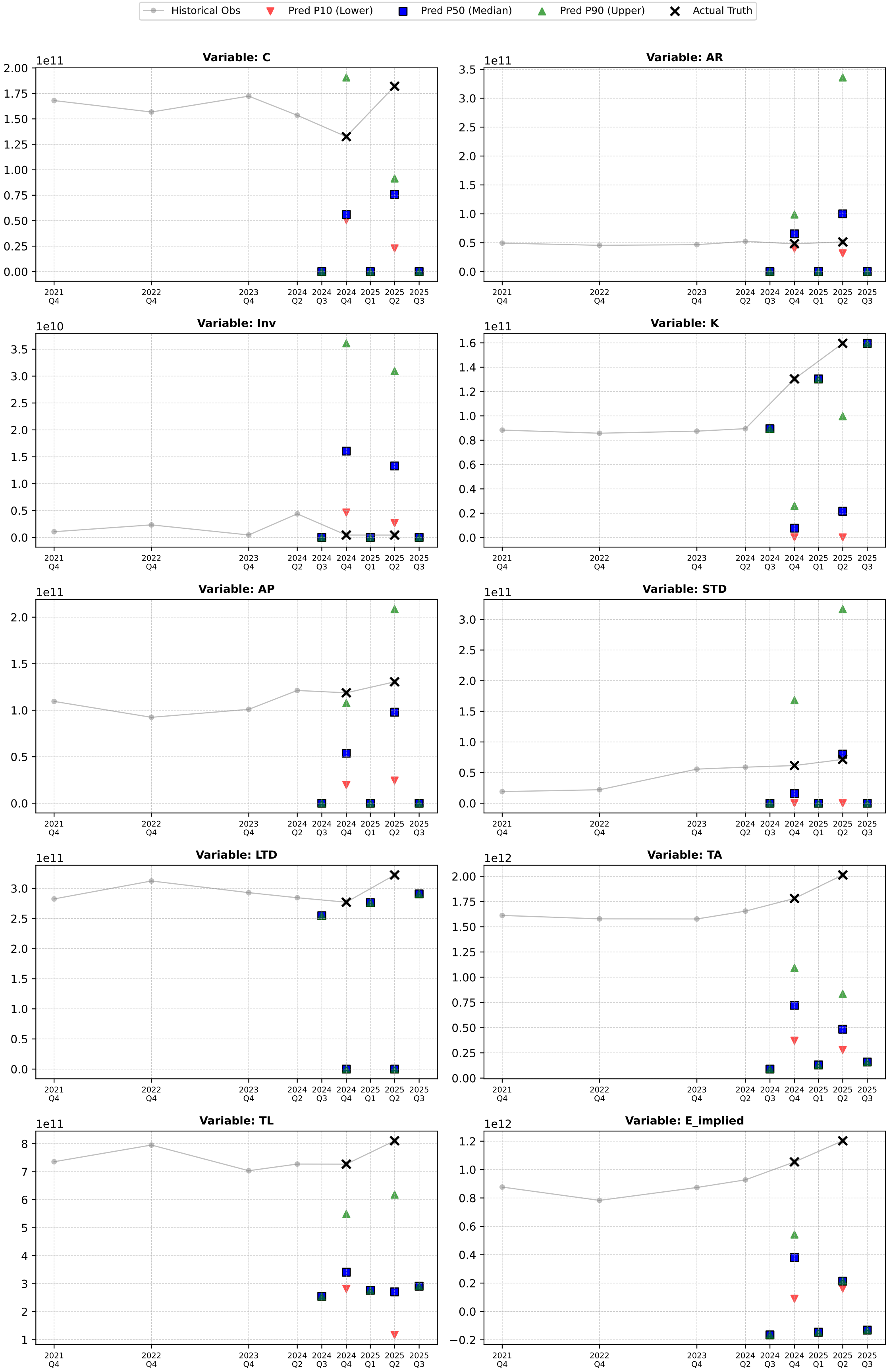

Figure 4.1: Prediction Results for Tencent (0700.HK): Historical Observations vs. Actual Truth (X) vs. Predicted Quantiles (P10, P50, P90)

- **Handling of Non-linear Spikes:** A notable challenge in recursive Markovian models is capturing abrupt financial shifts. For Tencent, the *Cash (C)* and *Net PPE (K)* accounts experienced significant spikes in 2025 Q2. While the actual values (black X) exceeded the P90 "optimistic" scenario in these specific periods, the model's P50 (median) forecast successfully tracked the general upward trend from 2024.

- **Precision in Debt Management:** Unlike traditional global/sector parameterization which often erroneously predicts a "collapse" of debt to zero, the ML-based hybrid model accurately maintains the scale of *Short-term Debt (STD)* and *Long-term Debt (LTD)*. The predicted quantiles remain stable and closely aligned with Tencent's actual leverage management.

- **Guaranteed Accounting Consistency:** Every forecasted state $(\hat{\mathbf{y}}^{(P10)}, \hat{\mathbf{y}}^{(P50)}, \hat{\mathbf{y}}^{(P90)})$ strictly adheres to the fundamental accounting identity $TA = TL + E$. As observed in the *Total Assets (TA)* and *E_implied* subplots, the structural constraints of the differentiable accounting layer ensure that even extreme quantile predictions are internally consistent and "no-plug".

4.9.2 Alibaba: 9988.HK (Consumer Discretionary)

Figure 4.2 illustrates the similar rolling quarterly forecast for Alibaba (9988.HK). As a leading entity in the Consumer Discretionary sector, its financial structure presents unique challenges for recursive time-series modeling.

- **Working Capital Divergence:** A significant observation is the "hallucination" in *Inventory (Inv)* and *Accounts Payable (AP)*. While the actual truth (black X) for these accounts remains near zero, the model's P50 and P90 quantiles predict significantly higher levels. This suggests that Alibaba’ s operational efficiency and asset-light strategy in certain segments deviate from the generalized parameters learned from the broader sector.

- **Debt Scale Underestimation:** Similar to the baseline findings, the model struggles to capture the full scale of *Long-term Debt (LTD)*. (Although 2025 Q2 seems very good.) The actual truth for LTD is consistently higher than the P90 "optimistic" bound in late 2025. This indicates a "smoothing bias" where the neural forecaster under-predicts the magnitude of Alibaba's strategic leverage increases.

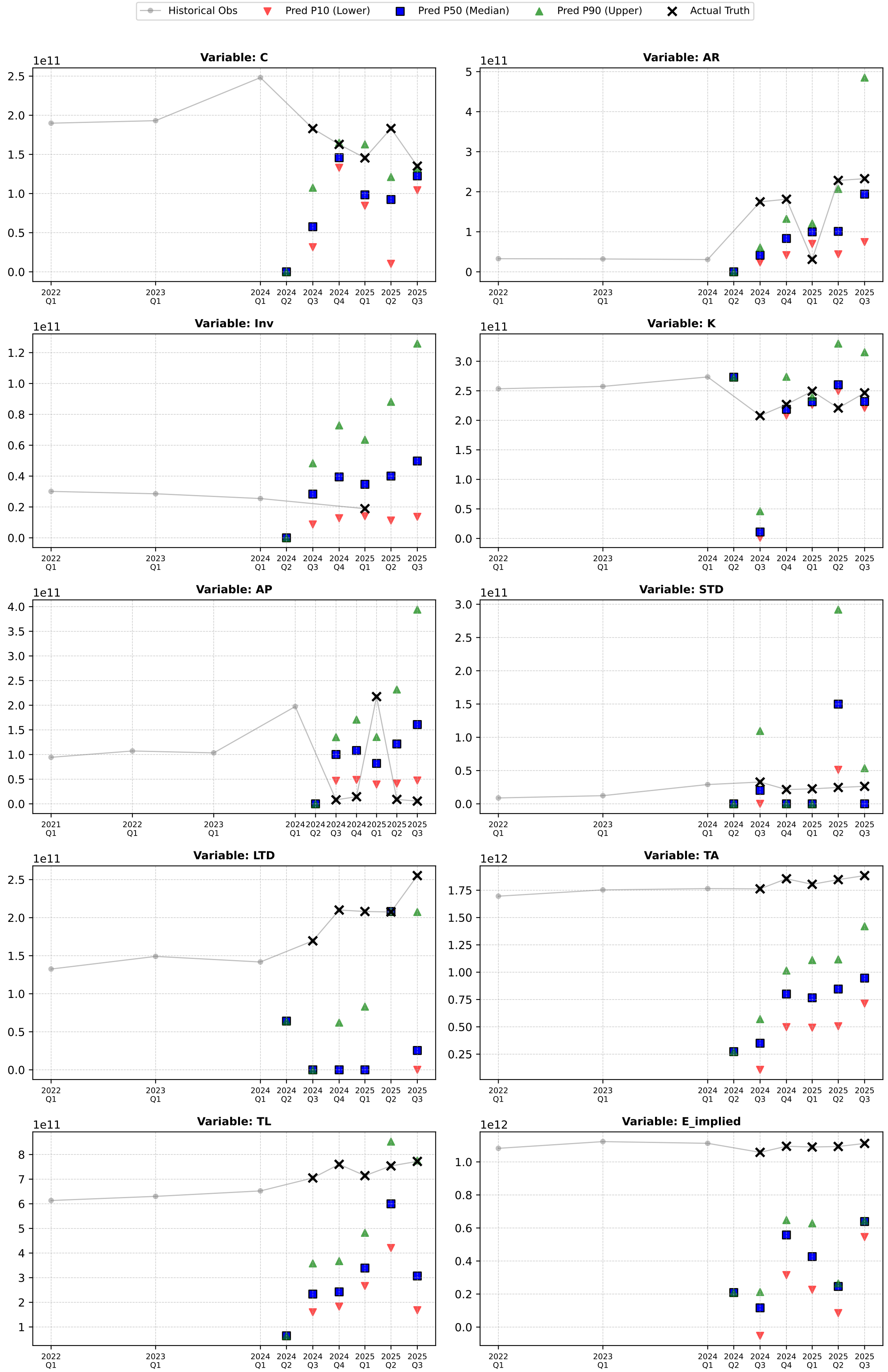

Figure 4.2: Prediction Results for Alibaba (9988.HK): Quantile Forecasts (P10, P50, P90) vs. Actual Truth (X)

- **Dynamic Cash Tracking:** For *Cash (C)*, the model (P50) successfully tracks the general direction of the downward trend from 2024 to early 2025, though it remains conservative relative to the actual truth' s volatility. The P10-P90 range effectively covers the truth for several mid-period steps, providing a valid probabilistic buffer for liquidity risk assessment.

- **Identity Preservation:** Despite the discrepancies in individual account levels, the differentiable accounting layer ensures that the $TA = TL + E$ identity is strictly preserved at all quantile levels. This structural consistency is visible in the *Total Assets (TA)* and *Total Liabilities (TL)* subplots, where the forecasted growth trends remain logically synchronized.

4.9.3 Alphabet Inc.: GOOG (Communication Services)

Figure 4.3 displays the rolling forecast results for Alphabet (GOOG). The performance of the Hybrid TFT-Accounting model on this mega-cap tech entity highlights the contrast between steady capital investment and highly efficient, sector-defying working capital management.

- **High Fidelity in Capital Modeling:** The model demonstrates exceptional accuracy in tracking *Net PPE (K)*. The actual truth (black X) aligns almost perfectly with the P50 median forecast throughout the 2024–2025 period. This indicates that the neural forecaster has successfully captured the steady, massive-scale infrastructure investment characteristic of Alphabet' s business model.

- **Liquidity and Working Capital "Hallucinations":** Similar to observations for other industry leaders, the model exhibits an upward bias in *Accounts Receivable (AR)*, *Accounts Payable (AP)*, and *Cash (C)*. While the actual values remain relatively flat and highly efficient, the P50 and P90 quantiles predict higher levels. This suggests that Alphabet' s internal cash management and collection efficiency are superior to the "average" sector-level behaviors learned by the model.

- **Conservative Debt Utilization:** The actual truth for both *Short-term Debt (STD)* and *Long-term Debt (LTD)* remains near zero, reflecting Alphabet's low-leverage capital structure. While the model predicts a non-zero debt trajectory (P50), the actual truth often sits near or below the P10 downside scenario, highlighting a deviation from standard financing hierarchies predicted by sector-wide data.

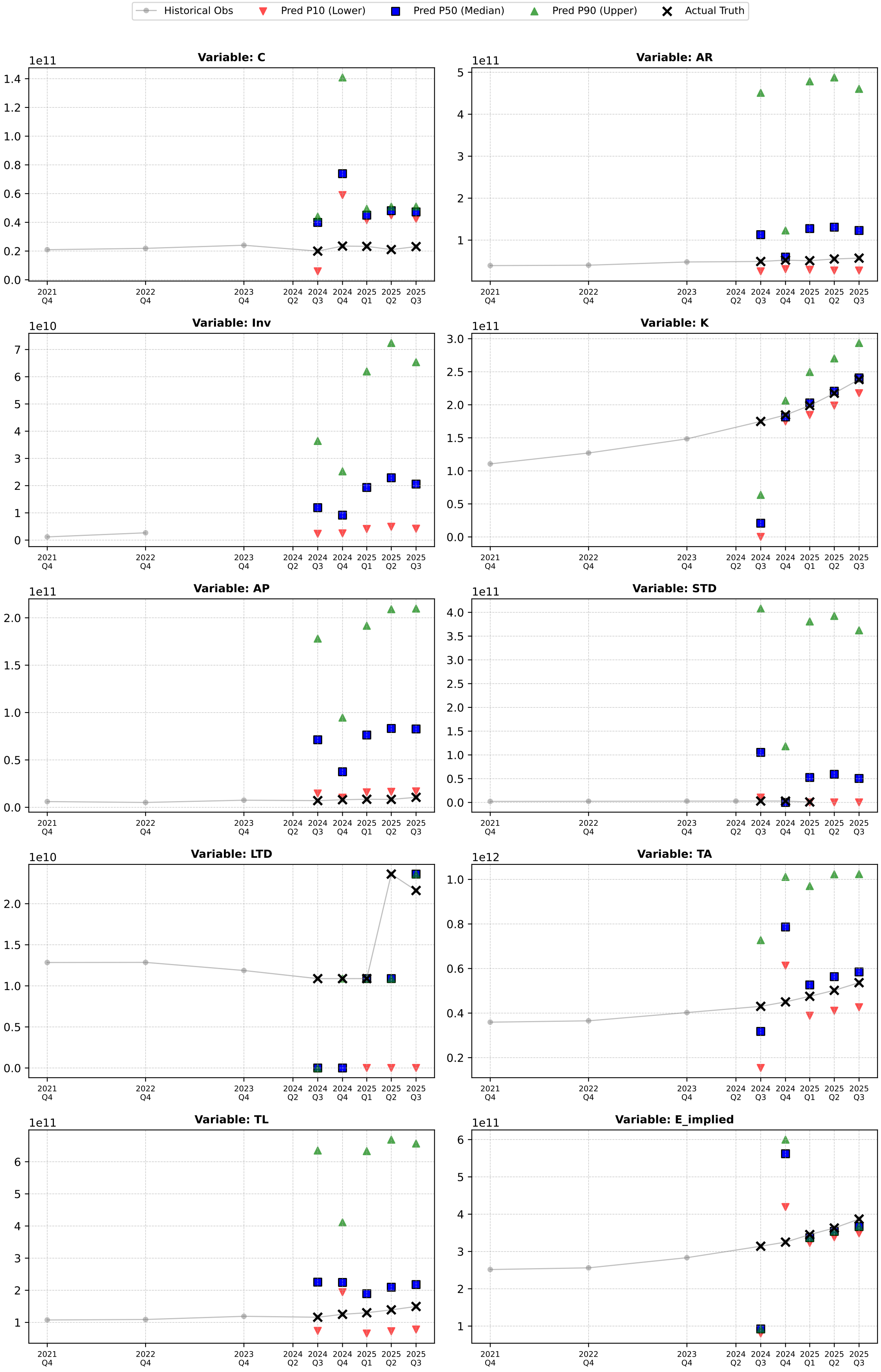

Figure 4.3: Prediction Results for Alphabet (GOOG): Quantile Forecasts (P10, P50, P90) vs. Actual Truth (X)

- **Structural Coherence:** Despite the magnitude differences in liquidity accounts, the *Total Assets (TA)* and *E_implied* subplots show that the model captures the overall growth trend of the company. The differentiable accounting layer maintains strict $TA = TL + E$ consistency, ensuring that the forecasted expansion of the balance sheet remains structurally sound.

4.9.4 JPMorgan Chase: JPM (Financials)

The rolling forecast for JPMorgan Chase (JPM) is presented in Figure 4.4. As a major financial institution, JPM represents a significant departure from the non-financial corporate models, testing the hybrid model's adaptability to the banking sector's unique balance sheet structure.

- **Industry-Specific Zero Inventory:** Consistent with the operational nature of a financial institution, the *Inventory (Inv)* subplot correctly shows a zero value for both actual truth (X) and all predicted quantiles. This confirms the model's ability to handle sectoral specificities where certain assets naturally collapse to zero.

- **Significant Scale Mismatch in Assets and Liabilities:** A critical failure is observed in the *Total Assets (TA)* and *Total Liabilities (TL)* subplots. The actual values (black X) remain in the magnitude of 10^{12}, while the model predicts a near-total collapse toward zero (10^{11} scale). This demonstrates that generalized parameters learned from non-financial firms are entirely insufficient for capturing the scale and structural stability of a global bank's balance sheet.

- **Divergence in Debt and Cash Management:** The model significantly under-predicts *Long-term Debt (LTD)* and *Cash (C)*, while over-predicting *Accounts Receivable (AR)*. These discrepancies highlight the "Financial Sector Divide," where the credit and liquidity management logic of a bank does not align with the turnover-based dynamics of industrial firms.

- **Structural Integrity via Differentiable Layer:** Despite the extreme numerical deviations in scale, the differentiable accounting layer still enforces the $TA = TL + E$ identity. However, due to the magnitude errors in assets and debt, the *E_implied* forecast exhibits high volatility and even enters negative territory in certain quantile scenarios, reflecting the model's struggle to balance a bank's massive accounts.

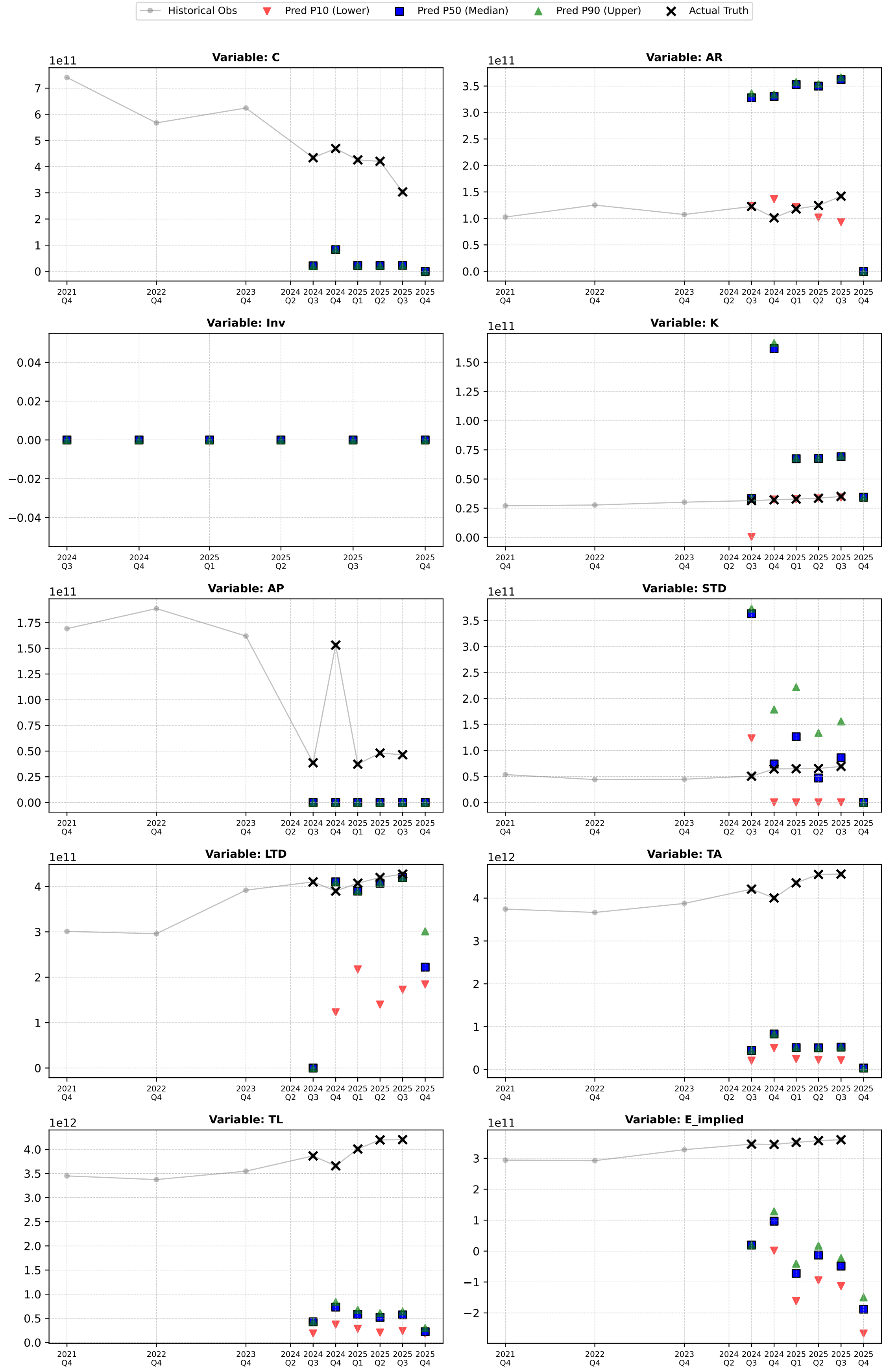

Figure 4.4: Prediction Results for JPMorgan Chase (JPM): Quantile Forecasts (P10, P50, P90) vs. Actual Truth (X)

4.9.5 Microsoft Corporation: MSFT (Information Technology)

Figure 4.5 presents the rolling forecast results for Microsoft (MSFT). As a dominant force in the Information Technology sector, Microsoft's balance sheet is characterized by massive capital investment in cloud infrastructure and strong equity growth, providing a robust test for the model's trend-tracking capabilities.

Experimental Observations:

- **Precise Tracking of Growth Drivers:** The model demonstrates exceptional fidelity in tracking *Net PPE (K)* and *Implied Equity (E_implied)*. For both accounts, the actual truth (black X) stays closely aligned with the P50 median forecast, effectively capturing Microsoft's sustained infrastructure expansion and profitability-driven equity accumulation.

- **Efficiency-Driven "Hallucinations":** Similar to other tech giants, the model tends to over-predict *Inventory (Inv)* and *Accounts Receivable (AR)*. While Microsoft maintains extremely lean inventory and efficient collections (black X remaining near historical lows), the model predicts higher levels based on sector-wide turnover averages. This reinforces the need for company-specific fine-tuning in highly efficient firms.

- **Debt and Liability Stability:** The model accurately captures the scale of *Long-term Debt (LTD)*, with actual values falling well within the P10-P90 quantile range. For *Short-term Debt (STD)*, the actual values remain near zero, which is mostly covered by the lower quantile (P10) of the model's forecast.

- **Structural Reliability:** The *Total Assets (TA)* and *Total Liabilities (TL)* subplots exhibit a clear upward trend in both actual and predicted data. The differentiable accounting layer ensures that despite individual account biases, the overall balance sheet remains internally consistent, adhering to the fundamental identity $TA = TL + E$.

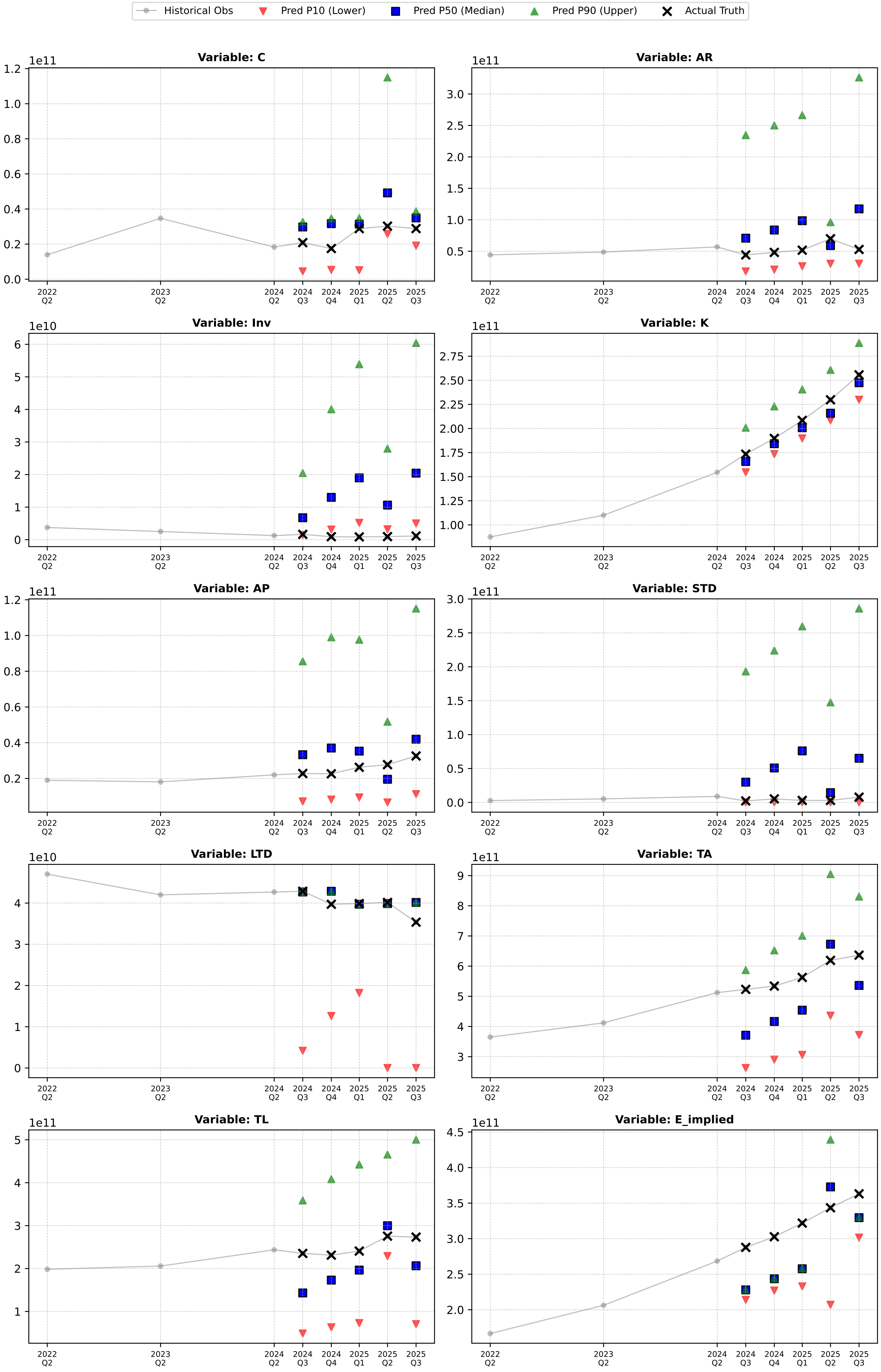

Figure 4.5: Prediction Results for Microsoft (MSFT): Quantile Forecasts (P10, P50, P90) vs. Actual Truth (X)

4.9.6 Volkswagen ADR: VWAGY (Consumer Discretionary)

Figure 4.6 presents the rolling backtest results for Volkswagen AG (VWAGY). As a capital-intensive manufacturing leader, Volkswagen provides a critical test case for the model's ability to track complex PPE dynamics and large-scale debt structures within the automotive industry.

- **High Precision in Capital Investment Tracking:** The model demonstrates remarkable accuracy in tracking *Net PPE (K)*. The actual truth (black X) follows the P50 median forecast very closely as it trends upward from 2024 to 2025. This indicates that the neural forecaster successfully internalized the depreciation and investment cycles typical of a global automotive manufacturer.

- **Stability of Working Capital vs. Sector Trends:** While the model predicts an upward trend in *Accounts Receivable (AR)* and *Accounts Payable (AP)* based on broader sector heuristics, Volkswagen's actual truth remains significantly more stable. This discrepancy suggests that Volkswagen maintains higher operational efficiency in its supply chain management than the "average" sector-level parameters would imply.

- **Debt Scale and Leverage Modeling:** The model successfully captures the magnitude of *Long-term Debt (LTD)*, with the actual truth remaining within the P10-P90 probabilistic range. However, for *Short-term Debt (STD)*, the model predicts several volatility spikes in 2025 that did not materialize in the actual data, reflecting the difficulty of predicting tactical financing shifts.

- **Macro Consistency:** The *Total Assets (TA)* and *E_implied* forecasts align well with the actual scale of the company (in the 10^{12} and 10^{11} ranges respectively). The differentiable accounting layer ensures that even when specific accounts like AR or AP deviate, the fundamental balance sheet identity $TA = TL + E$ is preserved across all forecasted quantiles.

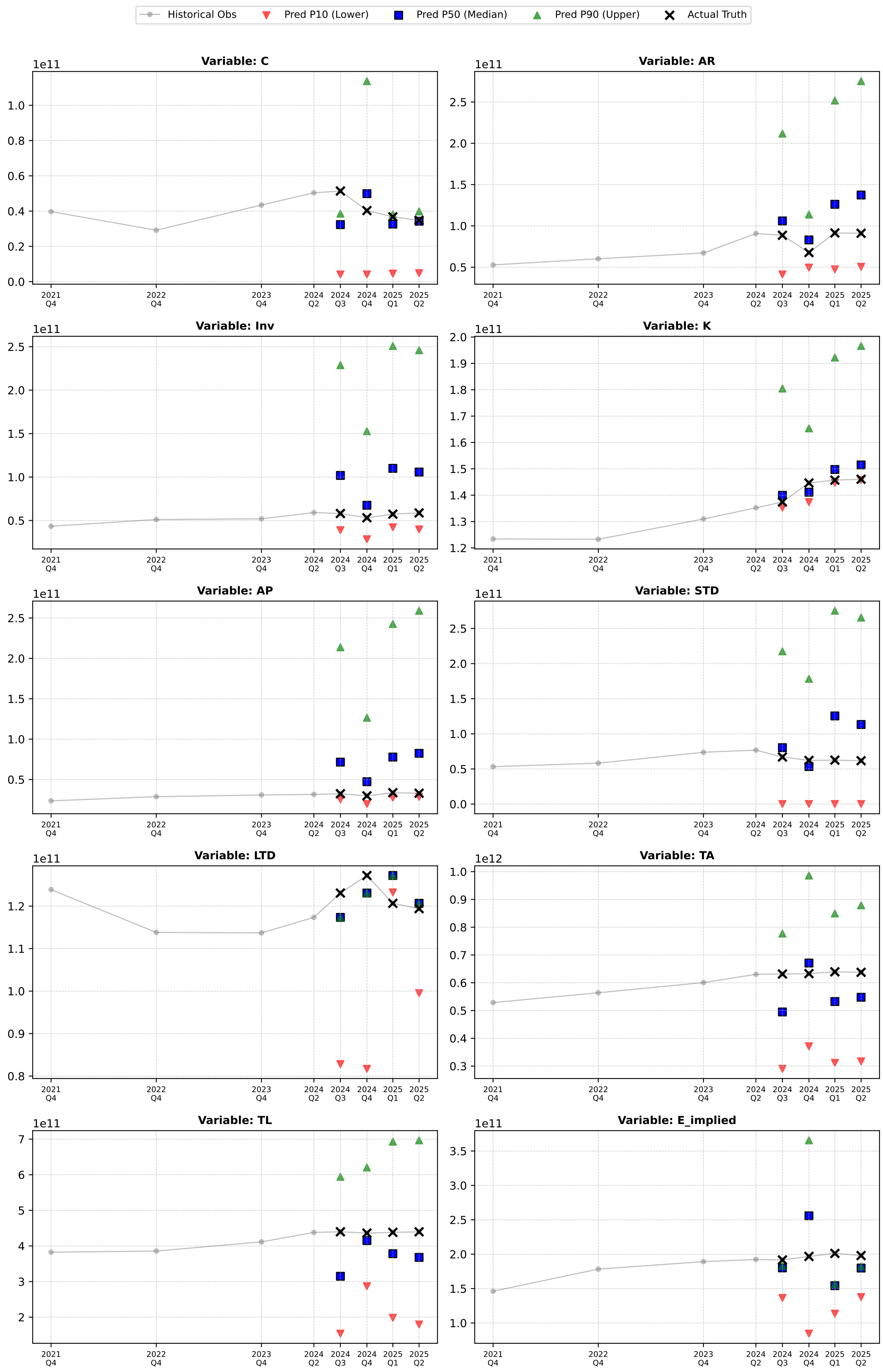

Figure 4.6: Prediction Results for Volkswagen ADR (VWAGY): Quantile Forecasts (P10, P50, P90) vs. Actual Truth (X)

4.9.7 Exxon Mobil Corporation: XOM (Energy)

Figure 4.7 illustrates the rolling forecast results for ExxonMobil (XOM). As a global leader in the Energy sector, XOM’ s balance sheet is heavily influenced by large-scale capital projects and commodity price cycles, providing a rigorous test for the model’ s ability to handle asset spikes and massive account scales.

- **Underestimation of the 2024 PPE Spike:** A critical observation in the *Net PPE (K)* subplot is the sharp step-change in actual values (black X) starting in early 2024. While the model (P50) correctly predicts an upward trajectory, it underestimates the magnitude of this jump, reflecting a "smoothing bias" common in recursive neural networks. However, the P90 optimistic bound provides a better risk coverage for this expansion than the deterministic baseline model.

- **Inventory and Working Capital Conservatism:** For *Inventory (Inv)* and *Accounts Receivable (AR)*, the actual data remains relatively stable. The model’ s P50 forecast exhibits an upward slope that slightly overshoots the truth, while the actual values reside near the P10 downside scenario. This suggests that XOM’ s inventory management is more lean and efficient than the sector-average heuristics learned by the TFT.

- **High Fidelity in Total Scale:** Despite the lag in capturing the exact timing of capital spikes, the model demonstrates excellent performance in tracking the overall magnitude of *Total Assets (TA)* and *Implied Equity (E_implied)*, both operating in the 10^{11} to 10^{12} range. The P10-P90 range effectively encapsulates the actual values for these macro-indicators throughout most of the test period.

- **Accounting Identity Preservation:** The differentiable accounting layer ensures that every forecasted step remains internally consistent. This is particularly evident in the synchronization between the *Total Assets (TA)* and *Total Liabilities (TL)* subplots, where the fundamental identity $TA = TL + E$ is strictly maintained even under high-uncertainty (P90) scenarios.

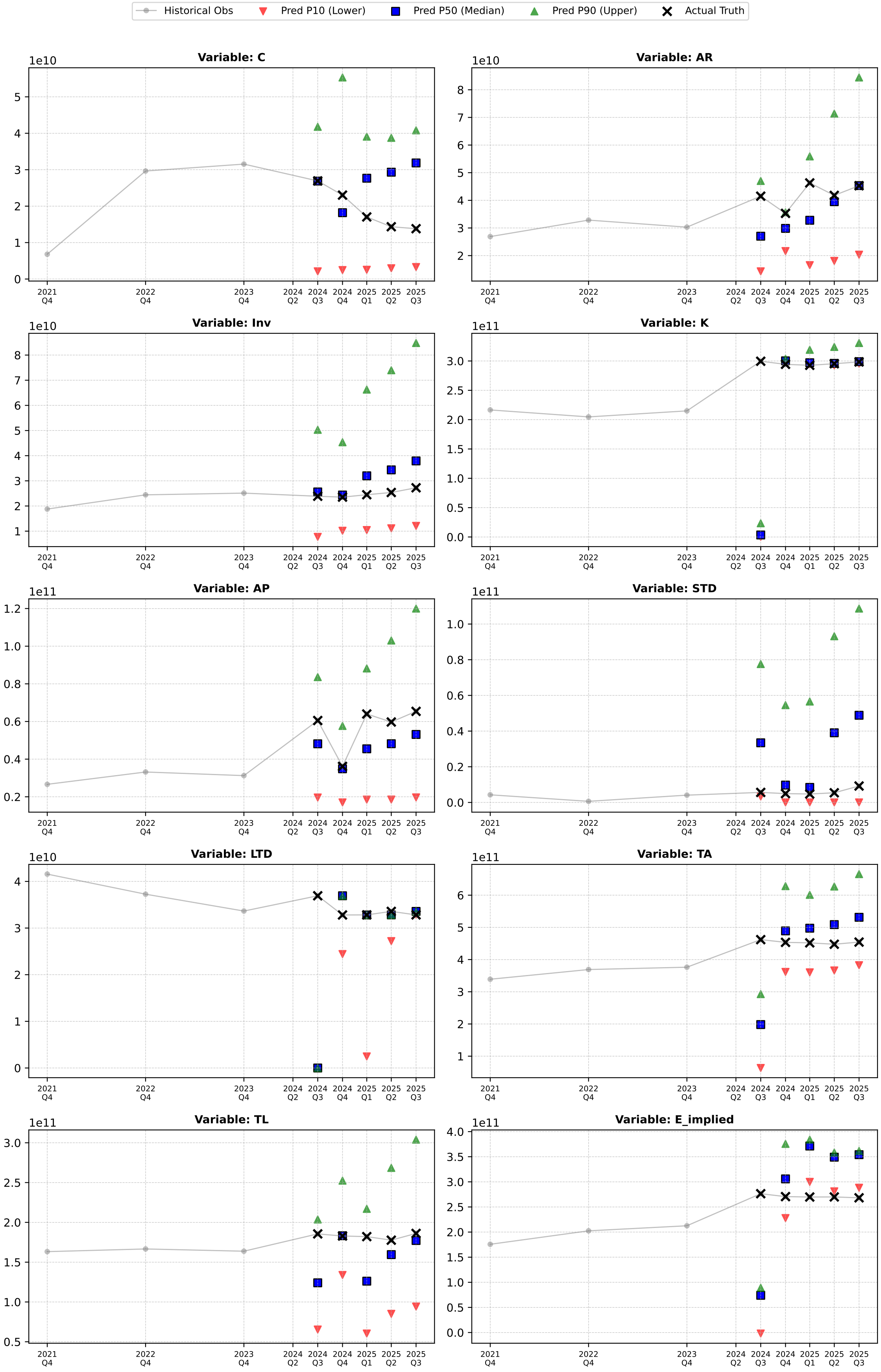

Figure 4.7: Prediction Results for Exxon Mobil Corporation (XOM): Quantile Forecasts (P10, P50, P90) vs. Actual Truth (X)

CHAPTER 5

LLM-based Balance Sheet Forecasting (Prompted Recursive Simulation)

5.1 Prompt Designing

Objective. We adopt the input-output framework established in Section 4, substituting the constrained Temporal Fusion Transformer (TFT) with a Large Language Model (LLM) designed for one-step-ahead forecasting. The LLM operates under explicit accounting constraints to ensure financial consistency:

$$\hat{y}_{t+1} = f_{\text{LLM}}(x_{t+1}, \{(x_\tau, y_\tau)\}_{\tau \leq t}, \mathscr{C}), \tag{5.1}$$

where x_t denotes the vectors of flow drivers and y_t represents the balance-sheet stocks.

Variable Definitions. The input features and target variables are represented as:

$$x_t = (X_S, X_{COGS}, X_{OPEX}, X_{EquityIssues}, X_{NI}, X_{Div}),$$

$$y_t = (C, AR, Inv, K, AP, STD, LTD, TA, TL, E_{\text{imp}}).$$

Hard Accounting Constraints. The prompt explicitly encodes fundamental accounting identities and physical feasibility:

$$E_{\text{imp}} = TA - TL \quad \text{(Identity; equity is a \textit{derived} residual)}, \tag{5.2}$$

$$C, AR, Inv, K, AP, STD, LTD, TA, TL \geq 0, \quad \text{(Non-negativity)}, \tag{5.3}$$

$$TA \geq C + AR + Inv + K, \tag{5.4}$$

$$TL \geq AP + STD + LTD. \tag{5.5}$$

Equations (5.4) and (5.5) allow for non-negative residual categories (e.g., “Other Assets/Liabilities”), ensuring the model satisfies aggregation without requiring arbitrary “plug” variables.

Soft Heuristics. Leveraging the historical "warm-start" data provided in the prompt, the LLM is instructed to extrapolate stable financial ratios (e.g., AR/X_S, AP/X_{COGS}, C/TA) while enforcing temporal smoothness across the forecast horizon. The implied equity E_{imp} is subsequently recomputed via (5.2).

Recursive Rollout. Following the observation of historical steps $t = 0, \ldots, L$, the LLM generates a one-step forecast for $t = L + 1$. This prediction is then appended to the input sequence, and the process is repeated autoregressively to generate multi-step simulations.

5.2 Model Selection and Implementation Logic

To evaluate the frontiers of LLM reasoning in financial contexts, we selected two state-of-the-art models: **GPT-5.2 Thinking** and **Gemini 3 Pro**.

Our baseline TFT model was trained on a comprehensive S&P 500 dataset (via `yfinance`), excluding seven specific tickers (GOOG, MSFT, JPM, XOM, VWAGY, Tencent, and Alibaba) reserved for out-of-sample testing. In contrast, we opted for a **zero-shot prompting approach** for the LLMs rather than providing specific training subsets. This decision is based on two primary considerations:

1. **Pre-trained Financial Knowledge:** These frontier models have been exposed to vast corpuses of corporate filings and financial statements during pre-training, granting them an inherent understanding of sector-specific dynamics.

2. **Data Density Limitations:** Current data acquisition via `yfinance` provides limited time-series depth (often fewer than 10 annual/quarterly data points per ticker). Such sparsity poses a risk for fine-tuning; while one could theoretically apply **Supervised Fine-Tuning (SFT)** with **LoRA (Low-Rank Adaptation)** to a smaller reasoning model (e.g., Qwen3-0.6B), the lack of diverse temporal patterns increases the likelihood of performance degradation or catastrophic forgetting during **Contrastive Pre-training (CPT)**.

Consequently, we implement prediction via **Prompted Recursive Simulation**, leveraging the models' in-context learning and symbolic reasoning capabilities to navigate the sparse data environment while adhering to the specified accounting rigors.

5.3 Prediction Result Comparison of Constrained TFT, GPT 5.2-Thinking and Gemini 3 Pro Models

5.3.1 Tencent (0700.HK): Rolling Forecast Results

Overall comparison. For Tencent, the LLM-based forecasts (GPT and Gemini) are consistently closer to the realized trajectories than the TFT model. A salient failure mode of TFT is the presence of pronounced forecast *spikes* in multiple accounts, while the realized series remain comparatively smooth. In contrast, both GPT and Gemini produce stable paths that better align with the observed dynamics, indicating stronger robustness under rolling (recursive) forecasting.

Account-level observations.

- **Cash (C):** TFT tends to overestimate the level; GPT/Gemini are generally more optimistic than the realized series, suggesting mild lag or level bias in late horizons.
- **Accounts Receivable (AR):** The realized AR is highly stable. GPT/Gemini track it very closely, whereas TFT exhibits an unrealistic upward spike that is inconsistent with the ground truth.
- **Inventory (Inv):** The realized inventory is near zero. All models overpredict, but TFT produces a much larger transient peak; LLMs remain biased upward yet materially closer to the realized values than TFT.
- **Fixed Assets (K):** The realized fixed assets increase steadily. TFT matches one period well but then flattens; GPT/Gemini systematically underestimate the upward trend, consistent with underestimating capital intensity.
- **Accounts Payable (AP):** GPT/Gemini nearly overlap with the realized AP, while TFT again produces an upward spike, indicating excess sensitivity to input fluctuations.
- **Short-Term Debt (STD):** The realized STD stays very low. GPT/Gemini fit it almost perfectly; TFT shows a large and implausible deviation (spike).
- **Long-Term Debt (LTD):** The realized LTD rises toward the end of the horizon. TFT captures the earlier level but misses the subsequent increase; GPT/Gemini are more stable but slightly underpredict the terminal rise.

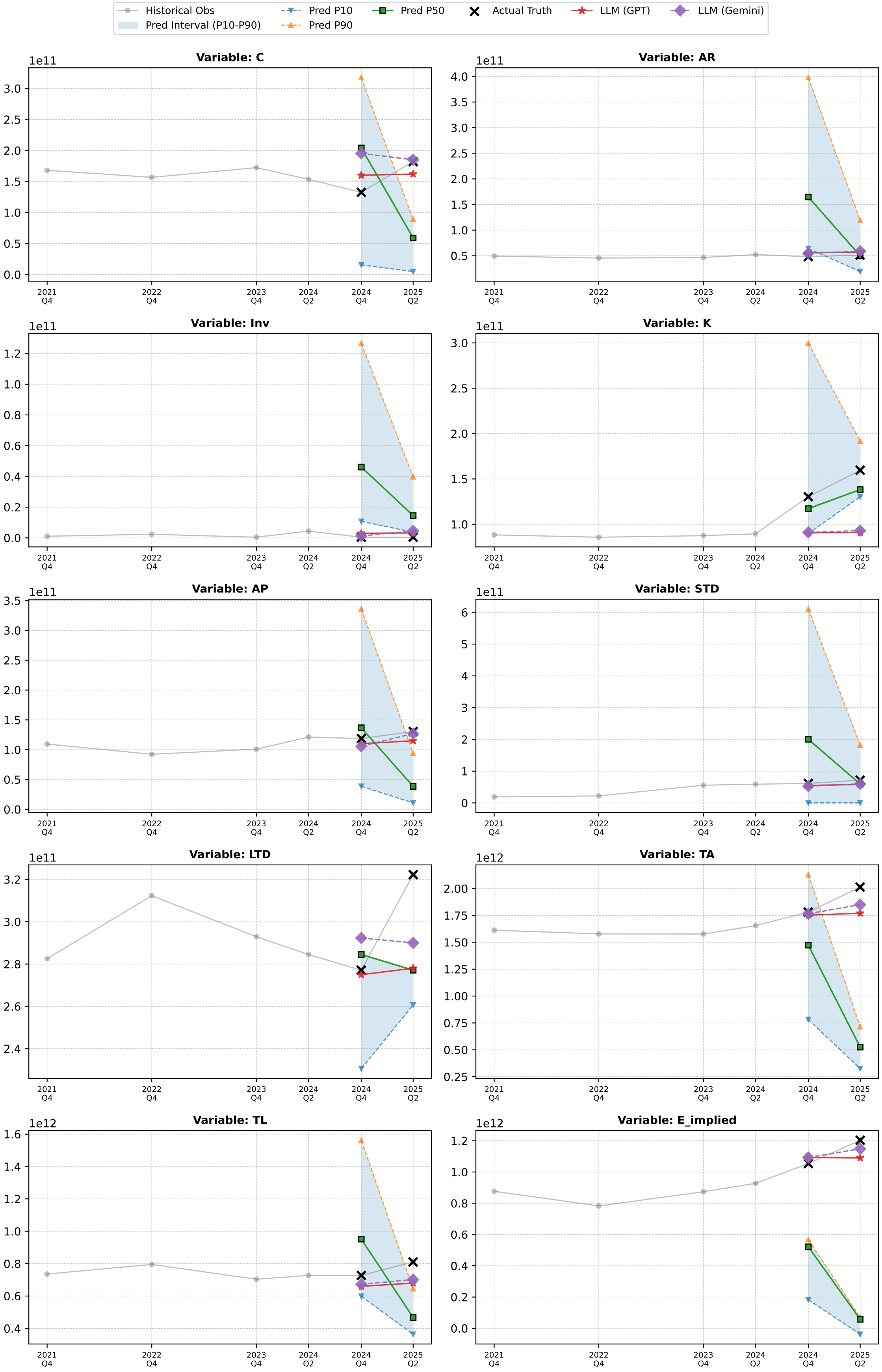

Figure 5.1: Rolling forecast results for Tencent (0700.HK): TFT quantile forecasts (P10, P50, P90) versus the realized truth (X), together with LLM point forecasts (GPT and Gemini) across key balance-sheet variables.

- **Total Assets (TA) and Total Liabilities (TL):** GPT/Gemini closely follow the growth pattern in TA (and the corresponding TL level), while TFT displays strong over-volatility and level underestimation in both aggregates.
- **Implied Equity (E_{imp}):** This is where the gap is most pronounced. LLM forecasts closely match the realized equity trajectory, whereas TFT can collapse toward unrealistically low values, indicating a breakdown in cross-account consistency under rolling prediction.

Interpretation. A plausible explanation is that TFT overreacts to noisy driver signals under limited quarterly samples (since the training data is limited), producing *over-volatility* that propagates through recursive rollouts. By contrast, GPT and Gemini may benefit from strong financial priors learned during pretraining, yielding more coherent and stable balance-sheet trajectories for large technology firms in sparse time-series settings.

5.3.2 Alibaba (9988.HK) Rolling Forecast Analysis

Overview. Figure 5.2 shows a highly polarized outcome. The TFT model exhibits a pronounced *spike syndrome*, with abrupt and implausible jumps concentrated around 2025 Q1 across many accounts, despite relatively smooth realized trajectories. In contrast, the LLM-based forecasts (GPT-5.2 and Gemini 3 Pro) behave more like a conservative auditing prior: they produce smooth, high-coherence paths that remain close to historical levels and the realized values.

Variable-wise performance.

- **Liquidity metrics (C, AR, Inv):** The realized series are broadly seasonal yet stable. TFT generates unrealistic surges (most visibly in AR and Inv) around 2025 Q1, whereas the LLMs preserve the historical scale and better match the realized levels.
- **Operational liabilities (AP, STD):** Alibaba’ s short-horizon liability structure appears stable in the ground truth. TFT predicts large, short-lived expansions (spikes) that are not supported by observations, while LLM forecasts remain near the realized trajectories with minimal deviation.

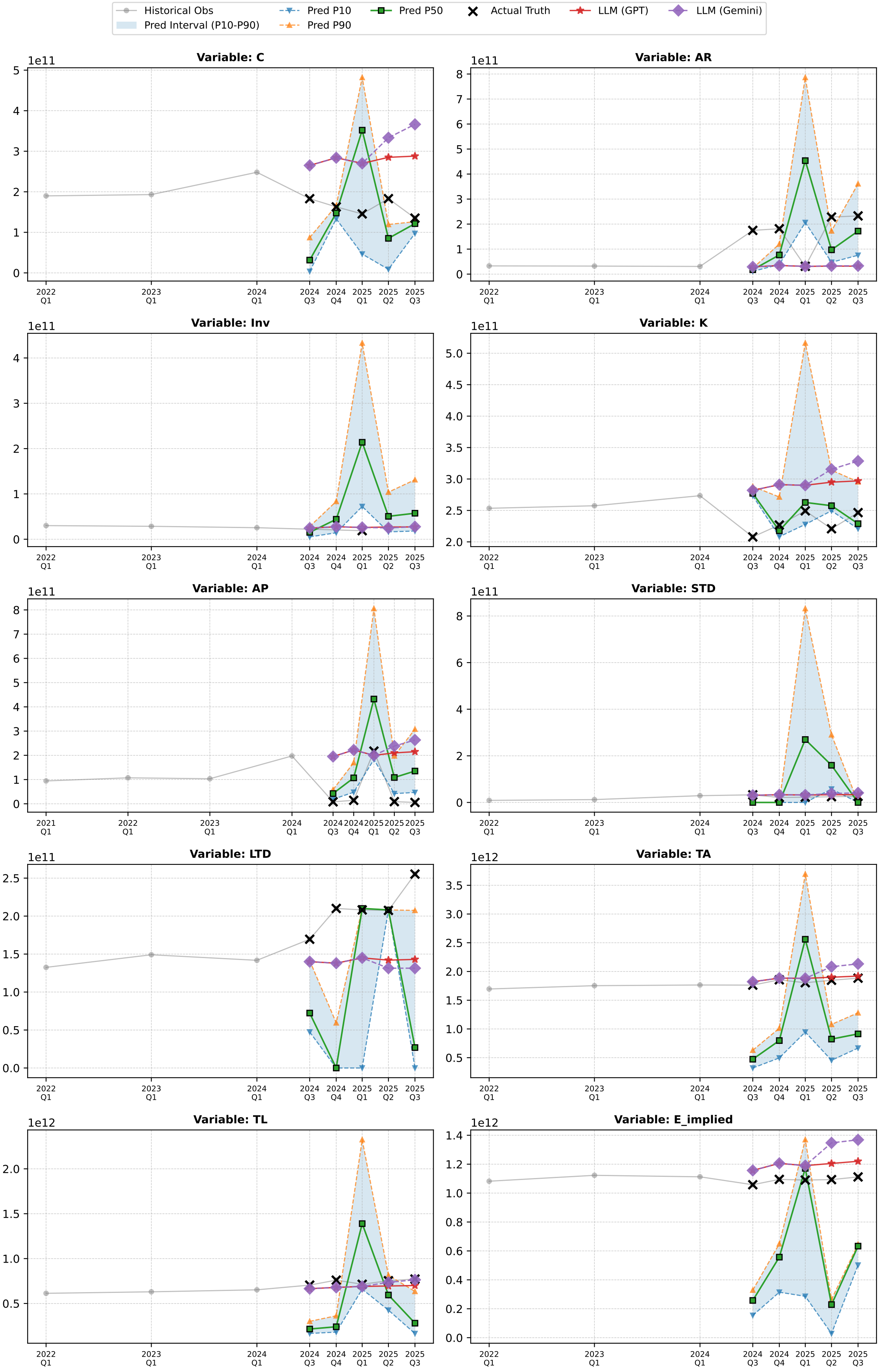

Figure 5.2: Rolling forecast results for Alibaba (9988.HK): TFT quantile forecasts (P10, P50, P90) versus the realized truth (X), together with LLM point forecasts (GPT-5.2 and Gemini 3 Pro) across key balance-sheet variables.

- **Capital structure (K, LTD):** The realized data suggest gradual capital/debt dynamics (with LTD increasing later in the horizon). LLMs may react slightly conservatively (mild lag in LTD), but still provide materially more interpretable forecasts than TFT’s high-amplitude oscillations.
- **Aggregate solvency (TA, TL, E_{imp}):** This is the most striking gap. LLMs closely track the growth patterns of TA and TL and maintain economically coherent implied equity. TFT, however, shows severe instability in aggregates and can produce equity trajectories that are inconsistent with the overall balance-sheet evolution under recursive rollout.

Conclusion. For Alibaba, the experiment suggests that TFT is prone to numerical divergence under multi-step recursive forecasting when trained on sparse quarterly samples, amplifying noise into extreme over-volatility. By contrast, LLM zero-shot reasoning, combined with explicit accounting-identity constraints, yields forecasts that are substantially more stable and commercially plausible for a mature mega-cap firm.

5.3.3 Google (GOOG) Rolling Forecast Analysis

Overview. For Google, the LLM-based forecasts again dominate in the presence of high-quality, low-noise financial dynamics. Gemini 3 Pro is particularly sensitive to long-horizon asset growth, while TFT continues to exhibit *numerical hallucinations* in liquidity-related accounts, producing spike-like deviations that are not supported by the realized series (Figure 5.3).

Variable-wise performance.

- **Operational stability (C, AR, AP, STD):** These operating accounts display strong inertia in the ground truth. TFT erroneously predicts a sharp expansion around 2024 Q4 (spikes in both level and uncertainty), whereas GPT/Gemini correctly preserve the stable scale and closely overlap with realized values.
- **Growth strategy (K):** The realized fixed assets reflect steady infrastructure investment. Gemini best matches the growth slope, GPT is slightly less accurate, and TFT is noticeably noisier with larger random perturbations.

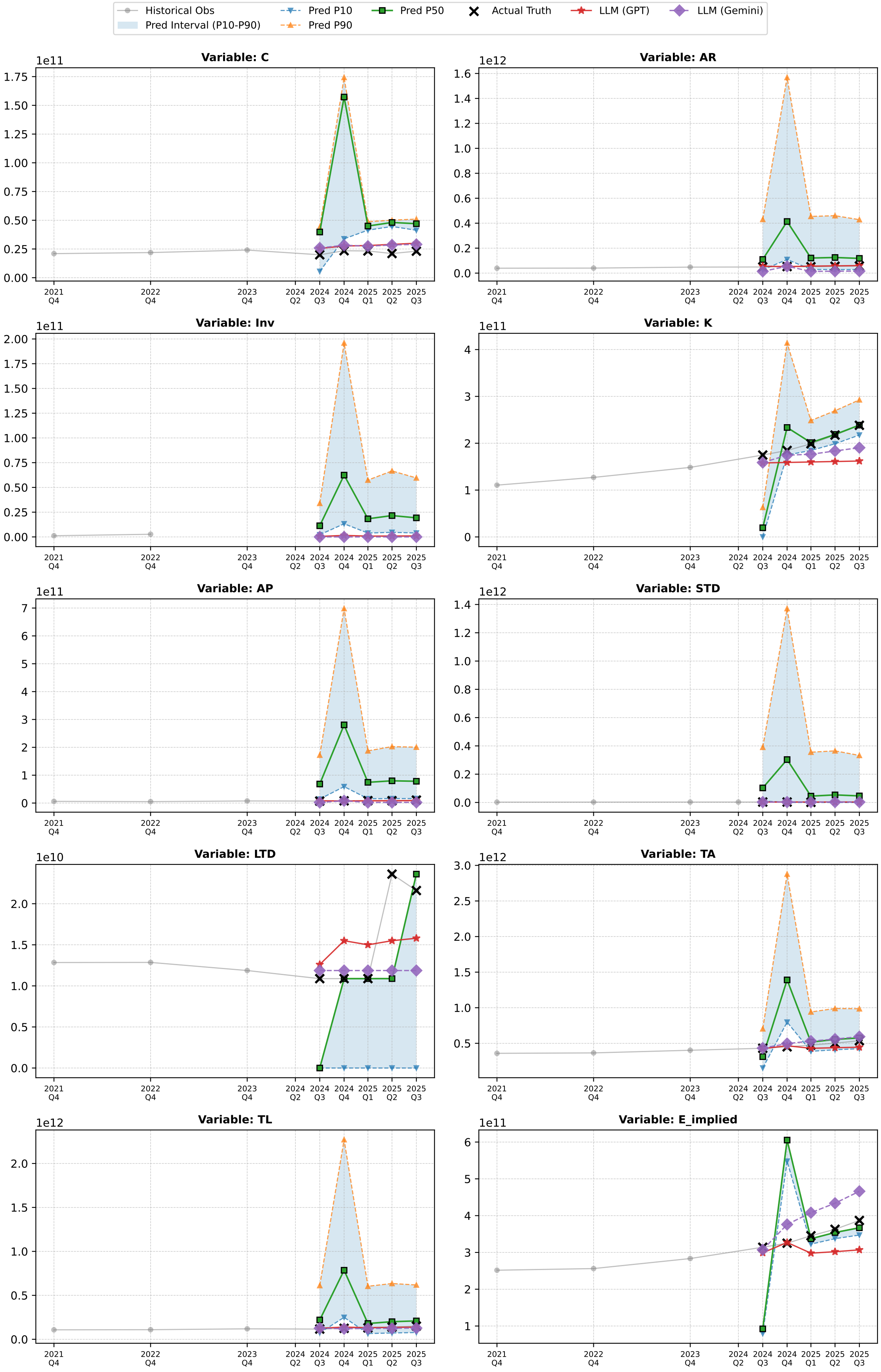

Figure 5.3: Rolling forecast results for Google (GOOG): TFT quantile forecasts (P10, P50, P90) versus the realized truth (X), together with LLM point forecasts (GPT-5.2 and Gemini 3 Pro) across key balance-sheet variables.

- **Financial anomaly (LTD):** A discontinuous jump in long-term debt around 2025 Q2 is not captured by either TFT or the LLMs, indicating that abrupt, management-driven financing decisions remain difficult to forecast from historical patterns alone.
- **Equity accrual (E_{imp}):** The persistent increase in implied equity is most accurately tracked by Gemini. While TFT's median may appear directionally reasonable at times, its wide P10–P90 band suggests low confidence in maintaining coherent balance-sheet evolution under recursive rollout.

Conclusion. This case reinforces that **Gemini exhibits stronger trend comprehension than GPT** for economically structured growth variables (e.g., capex-driven asset accumulation). However, all models face a common bottleneck when the target is a non-trending, discontinuous adjustment dominated by discretionary corporate actions (e.g., large debt issuance).

5.3.4 J.P. Morgan (JPM) Rolling Forecast Analysis

Overview. J.P. Morgan represents the most challenging setting in our study: a highly leveraged financial institution with a complex balance-sheet structure. In this case, the LLM-based forecasts (GPT-5.2 and Gemini 3 Pro) exhibit a clear and consistent advantage. The TFT model displays severe numerical instability under multi-step recursion, including near-zero collapse in several accounts and an "equity implosion" failure mode. By contrast, the LLM forecasts maintain balance-sheet coherence and produce economically plausible trajectories (Figure 5.4).

Variable-wise performance.

- **Monetary and operational assets (C, AR, Inv):** The realized series suggest disciplined liquidity management. GPT/Gemini track AR and Inv closely; for C, the LLMs tend to be biased high (level overestimation), yet remain substantially more stable than TFT, which exhibits post-2024 Q4 degradation and near-zero collapse.
- **Operational liabilities (AP):** The ground truth shows a pronounced jump around 2024 Q4. The LLMs capture this nonlinearity (a sharp upward adjustment followed by stabilization), while TFT largely misses the jump and instead produces an inconsistent trajectory.

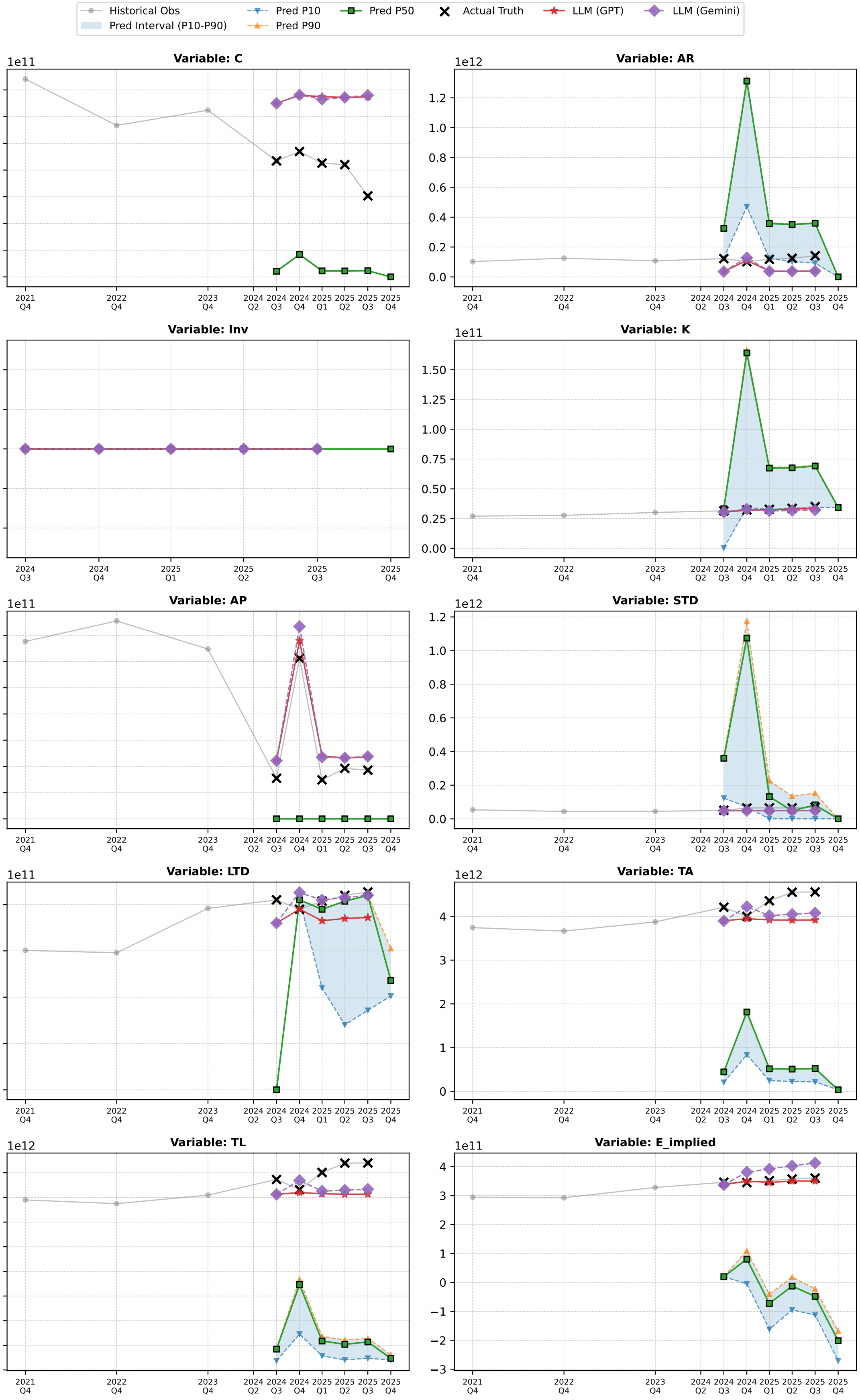

Figure 5.4: Rolling forecast results for J.P. Morgan (JPM): TFT quantile forecasts (P10, P50, P90) versus the realized truth (X), together with LLM point forecasts (GPT-5.2 and Gemini 3 Pro) across key balance-sheet variables.

]

- **Debt and fixed capital (STD, LTD, K):** LLM forecasts are accurate and stable for LTD and K. In contrast, TFT produces an extreme overshoot in STD (spike), consistent with an unstable density/quantile behavior under rare-event or heavy-tailed dynamics typical of bank balance sheets.
- **Aggregates (TA, TL, E_{imp}):** GPT/Gemini predictions for TA and TL nearly overlap with the realized points across the rolling horizon. The largest divergence occurs in implied equity: TFT drives E_{imp} toward implausibly low/negative values after 2024 Q4 (a "technical insolvency" implication), whereas LLM equity remains smooth and consistent with sustained capital adequacy.

Conclusion. The JPM case highlights a critical robustness limitation of TFT for high-leverage, tightly constrained financial sequences: recursive forecasting can amplify noise and produce logically inconsistent balance-sheet states. LLMs, leveraging strong pretrained financial priors and explicit accounting-identity constraints, better filter spurious fluctuations and deliver forecasts with substantially higher commercial interpretability.

5.3.5 Microsoft (MSFT) Rolling Forecast Analysis

Overview. For Microsoft, model performance forms a clear hierarchy: the LLM forecasts are superior to TFT in both stability and trend capture. A notable failure mode of TFT is a synchronized *numerical divergence* around 2025 Q2 across multiple accounts, producing implausible spikes under recursive rollout. In contrast, GPT and especially Gemini generate smooth, coherent trajectories that align well with Microsoft' s expansion narrative in the AI cycle (Figure 5.5).

Variable-wise performance.

- **Liquidity and operating items (C, AR, AP, STD):** The realized series indicate strong operational stability. TFT incorrectly predicts multi-fold expansions around 2025 Q2, whereas the LLM forecasts remain close to the realized points and preserve the historical scale.
- **Infrastructure expansion (K):** The ground truth reflects sustained investment in data centers and hardware assets. Gemini 3 Pro matches the growth slope remarkably well; GPT is slightly less accurate, while TFT exhibits excessive randomness and spike-like perturbations.

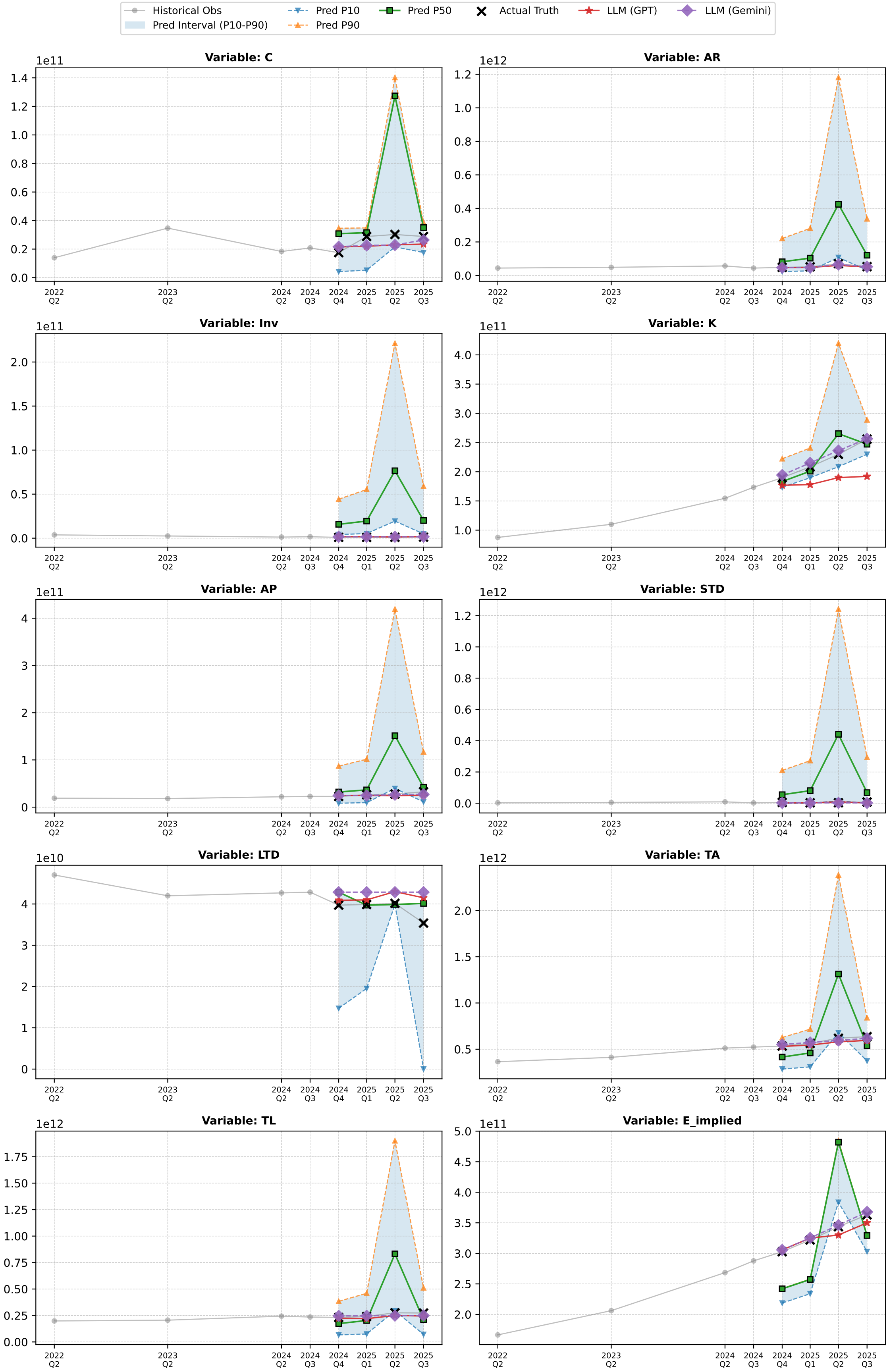

Figure 5.5: Rolling forecast results for Microsoft (MSFT): TFT quantile forecasts (P10, P50, P90) versus the realized truth (X), together with LLM point forecasts (GPT-5.2 and Gemini 3 Pro) across key balance-sheet variables.

- **Liability and solvency (LTD, TL):** Microsoft' s liability side is broadly stable in the realized data. The LLMs anchor to this stability, whereas TFT displays liability "hallucinations" that distort the implied capital structure in rolling forecasts.
- **Capital appreciation (E_{imp}):** Implied equity increases steadily, consistent with retained earnings accumulation. LLM trajectories (particularly Gemini) are accurate and internally coherent, while TFT fails to reconcile cross-account interactions under recursion, leading to economically implausible implied equity dynamics.

Conclusion. The MSFT case reinforces earlier findings: for large-cap firms with high-quality data and a clear growth mechanism, frontier LLMs can effectively substitute complex parametric models in zero-shot rolling forecasting. Gemini's sensitivity to infrastructure-driven asset expansion (notably K) makes it particularly strong for long-horizon value narratives.

5.3.6 Volkswagen (VWAGY) Rolling Forecast Analysis

Overview. Volkswagen provides a clear stress test for rolling (recursive) forecasting in a large, cyclical industrial firm. As shown in Figure 5.6, TFT experiences a systemic collapse characterized by widespread, implausible *spikes* around 2024 Q4 across most accounts, indicating strong sensitivity to noisy drivers under recursion. In contrast, the LLM forecasts (GPT-5.2 and Gemini 3 Pro) remain stable and economically coherent, closely tracking the comparatively smooth realized series.

Variable-wise performance.

- **Monetary and operational assets (C, AR, Inv):** The realized working-capital and liquidity accounts are relatively stable. TFT predicts multi-fold, short-lived expansions (spikes) at 2024 Q4 that are not supported by the observations, while the LLMs filter these disturbances and maintain the correct scale and smoothness.
- **Industrial infrastructure (K):** As a heavy-asset manufacturer, Volkswagen exhibits gradual capital dynamics. The LLMs capture this slow-moving pattern, whereas TFT injects pulse-like perturbations that reduce interpretability in a long-horizon rollout.

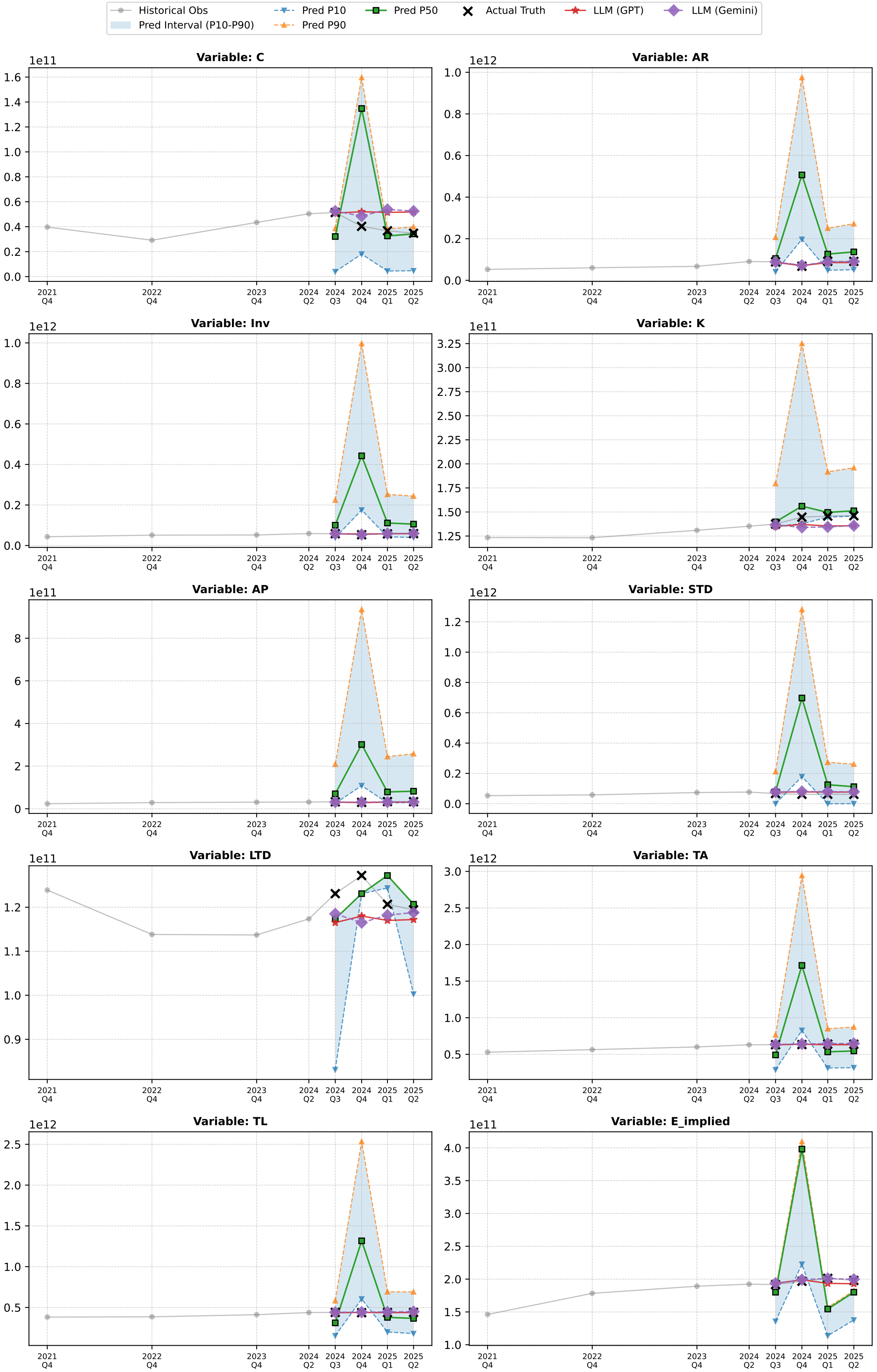

Figure 5.6: Rolling forecast results for Volkswagen (VWAGY): TFT quantile forecasts (P10, P50, P90) versus the realized truth (X), together with LLM point forecasts (GPT-5.2 and Gemini 3 Pro) across key balance-sheet variables.

- **Capital structure evolution (LTD):** The realized LTD displays a step-like upward shift later in the horizon. LLM forecasts reflect this nonlinear adjustment more faithfully than TFT, suggesting superior robustness to regime-like changes in financing structure.

- **Balance-sheet integrity (TA, TL, E_{imp}):** Despite severe volatility in TFT sub-accounts, the LLMs preserve accurate aggregate trajectories and produce a stable, well-anchored implied equity path. TFT, by contrast, oscillates between extreme expansions and contractions, undermining the usefulness of E_{imp} under recursive forecasting.

Conclusion. For a cyclical, large-scale industrial balance sheet with relatively smooth historical dynamics, LLM zero-shot forecasting (under accounting-identity constraints) is markedly more robust than TFT. Gemini's responsiveness to the long-term debt adjustment further suggests strong potential for capital-structure forecasting in heavy-industry settings.

5.3.7 ExxonMobil (XOM) Rolling Forecast Analysis

Overview. For ExxonMobil, TFT again suffers a systemic *spike failure* around 2024 Q4, producing implausible surges across multiple accounts under recursive rollout. By contrast, the LLM forecasts (GPT-5.2 and Gemini 3 Pro) remain markedly stable and better aligned with the relatively smooth realized trajectories typical of a large, mature energy firm (Figure 5.7).

Variable-wise performance.

- **Working-capital metrics (C, AR, Inv, AP, STD):** The realized series are largely stable over the evaluation window. TFT generates a synchronized 2024 Q4 spike across these accounts (with very wide P10–P90 bands), whereas GPT/Gemini closely track the smooth levels with small deviations.

- **Capital assets (K):** The realized K indicates a sizable scale expansion (increasing to the 10^{11}–10^{12} order shown in the panel). The LLMs correctly identify the upward direction but appear conservative in magnitude, while TFT injects pulse-like distortions consistent with its spike pathology.

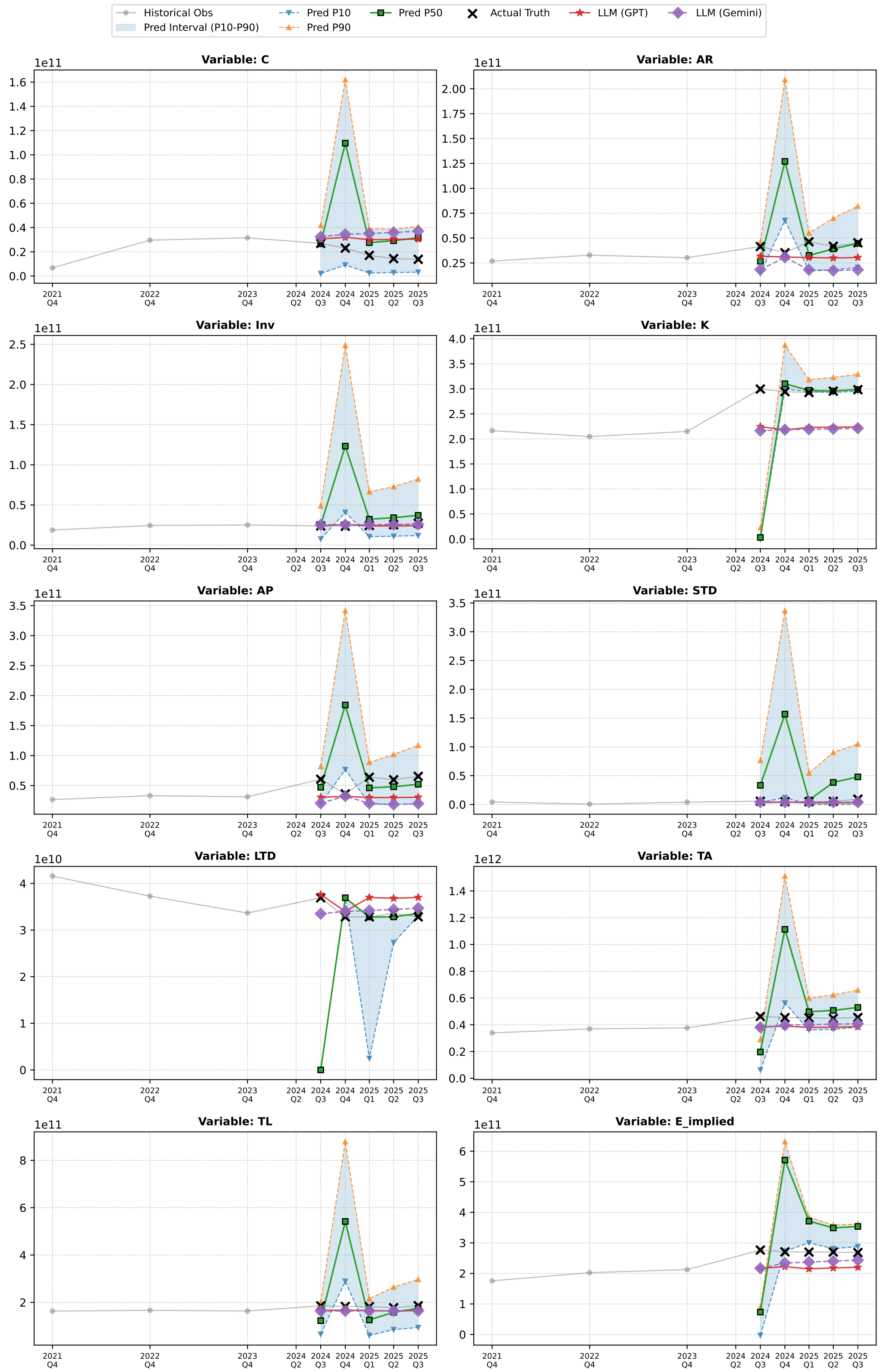

Figure 5.7: Rolling forecast results for ExxonMobil (XOM): TFT quantile forecasts (P10, P50, P90) versus the realized truth (X), together with LLM point forecasts (GPT-5.2 and Gemini 3 Pro) across key balance-sheet variables.

- **Debt management (LTD):** Long-term debt is comparatively stable in the realized data. GPT/Gemini remain tightly anchored to the observed level, whereas TFT deviates substantially and exhibits high volatility, reducing interpretability in a rolling setting.

- **Aggregate consistency (TA, TL, E_{imp}):** LLM forecasts preserve coherent aggregate evolution and implied equity trajectories that stay close to the realized points. TFT’ s failures in multiple components propagate to aggregates, yielding unstable TA/TL and an implied equity path that becomes economically unreliable under recursion.

Conclusion. The XOM case further supports the core finding of this report: for firms with strong industry structure and relatively stable accounting dynamics, frontier LLMs (under explicit accounting-identity constraints) outperform traditional probabilistic forecasting models by filtering short-horizon noise and maintaining long-run balance-sheet coherence.

5.4 Overall Summary and Discussion

Key takeaway. Across the seven case studies (GOOG, MSFT, JPM, XOM, VWAGY, 0700.HK, 9988.HK), the frontier LLMs (Gemini 3 Pro and GPT-5.2 Thinking) consistently produced smoother and more economically coherent rolling forecasts than TFT, especially for aggregate accounts and implied equity. Importantly, this does *not* imply that TFT is intrinsically inadequate: under the same limited quarterly data regime, TFT already represents a substantial improvement over purely theory-based or rules-driven forecasting approaches, providing a learnable mapping from drivers to financial statements and delivering distributional outputs (quantiles) rather than only point estimates.

Why TFT underperforms here: data limitations and domain shift. The performance gap observed in our experiments is more plausibly explained by a combination of (i) insufficient temporal coverage and (ii) a mismatch between the training distribution and the target firms:

- **Temporal scarcity.** Balance-sheet dynamics are slow-moving and regime-dependent; training and rolling prediction with only a short historical window makes TFT vulnerable to error accumulation under recursive rollouts. This can manifest as “spike” behavior and instability in derived aggregates (notably E_{imp}).

- **Firm specificity / distribution shift.** The seven evaluation tickers are atypical and structurally heterogeneous: mega-cap technology firms (GOOG, MSFT), a highly leveraged global bank (JPM), a mature energy major (XOM), a cyclical heavy-industry manufacturer (VWAGY), and two China-based mega-caps reported under different market conventions (0700.HK, 9988.HK). A dataset constructed broadly from S&P 500-style financials can be a weak proxy for these firms' idiosyncratic capital structure, working-capital regimes, and reporting patterns. In such settings, a generic TFT may learn an "average" mapping that is not well calibrated for these targets.

Why frontier LLMs look strong in this setting. Gemini 3 Pro and GPT-5.2 Thinking likely benefit from strong priors acquired during pretraining on large-scale financial text and tabular content (e.g., reports, filings, commentary, and related corpora). When combined with explicit accounting-identity constraints in the prompt, these priors translate into stable rollouts that preserve cross-account coherence and suppress noise amplification. In other words, the LLM behaves less like a purely statistical extrapolator and more like a constraint-aware analyst that prefers conservative, internally consistent balance-sheet evolution.

What would make TFT more competitive: better data and richer inputs. The most valuable next step is not architectural complexity but *information*:

- **Longer firm histories.** Expanding each target firm's historical time span (and, if feasible, higher-frequency statements) would directly improve temporal generalization and reduce recursive instability.
- **Event-aware inputs.** Many large discontinuities (e.g., sudden debt issuance, acquisitions, restructuring, regulatory shocks, commodity cycles, or AI-driven capex waves) are not inferable from past balance-sheet values alone. Encoding key corporate events as additional tokens/features (e.g., event type, timing, magnitude proxies) could substantially reduce "surprise" jumps such as abrupt changes in LTD.
- **Targeted adaptation.** Sector- or firm-conditioned models (e.g., hierarchical TFT, lightweight fine-tuning, or mixture-of-experts by industry) can address distribution shift while retaining TFT's advantage in calibrated uncertainty estimation.

Practical implication: complementary strengths suggest a hybrid path. From a lending and credit perspective, TFT and LLMs offer complementary value: TFT provides an explicit probabilistic forecast (quantiles and intervals) that is useful for risk assessment, while LLMs provide strong structural priors and cross-account coherence under sparse data. A principled ensemble—e.g., TFT for uncertainty quantification and an LLM for constraint-aware anchoring or regime/event interpretation—is a promising direction to improve both accuracy and robustness in balance-sheet forecasting for strategic lending decisions.

CHAPTER 6

PDF Annual Report Analyst (Gemini 3 Pro-based)

6.1 Objective and Problem Setting

The goal of this part is to build an *automatic* analyst tool that takes an annual report PDF as input, extracts (i) the income statement and (ii) the balance sheet, and then computes a standardized set of lending-relevant ratios, including liquidity, leverage, and coverage measures.

Concretely, given the extracted statements for the *current year*, the tool is required to answer:

i) Net income.

ii) Cost-to-income ratio.

iii) Quick ratio, debt-to-equity, debt-to-assets, debt-to-capital, debt-to-EBITDA.

iv) Interest coverage ratio.

This chapter documents an end-to-end implementation — `annual_report_pipeline_Gemini.py` — which combines PDF parsing (`pypdf`), a structured LLM extraction step (Gemini API), deterministic ratio computation in Python, and a final LLM-based narrative layer.

6.2 System Overview

6.2.1 Inputs and Outputs

Input: a company annual report PDF.

Outputs:

- `tokens.md`: a tokenized representation of extracted statement values (canonical fields), plus explicit accounting constraints and diagnostic residuals.
- `answer.md`: an LLM-written response that reports the ratios and provides CFO/CEO recommendations, grounded on `tokens.md` and Python-computed metrics.

6.2.2 Pipeline Stages

The pipeline consists of seven stages:

1. **Locate** income statement / balance sheet / (optional) cash flow pages via a cheap text heuristic.
2. **Slice** the PDF to the identified statement pages only.
3. **Upload** the sliced PDF to the Gemini File API.
4. **Extract** all line-items and multi-year values using a JSON-only LLM prompt.
5. **Canonicalize** heterogeneous line-item labels into a fixed schema of fields.
6. **Compute** ratios deterministically in Python, and generate constraint diagnostics.
7. **Summarize** results and recommendations using a second LLM call.

6.3 Statement Page Localization

Because annual reports can exceed 100 pages, the script first identifies the statement pages before invoking the LLM. It scans at most the first 300 pages and scores each page with keyword weights plus a numeric-density term.

6.3.1 Keyword-weight scoring

For each page text T, a score is computed as

$$\text{score}(T) = \sum_{k\in\mathscr{K}} w_k \cdot \mathbf{1}[k \subset T] + \min\left(\left\lfloor \frac{N_{\text{num}}(T)}{40} \right\rfloor, 6\right), \tag{6.1}$$

where $\mathscr{K}$ is a keyword set for the statement type and $N_{\text{num}}(T)$ is the count of numeric-like strings (currency/parentheses/comma formats). The keyword weights used in the implementation are shown in Listing 6.1.

Listing 6.1: Statement-page keyword scoring used by `page_score`.

```
# income
("income statement", 8), ("statements of income", 8),
("net income", 6), ("operating income", 4),
("revenue", 3), ("sales", 2)

# balance
("balance sheet", 8), ("balance sheets", 8),
("total assets", 6), ("total liabilities", 4),
("total equity", 3), ("current assets", 2)

# cashflow
("cash flows", 8), ("statements of cash flows", 8),
("operating activities", 4), ("depreciation", 2),
("amortization", 2)
```

The final selection returns the best-scoring page indices for income and balance statements, and includes cash flow only if the cash-flow score is at least 8.

6.4 LLM-Based Table Extraction

6.4.1 PDF slicing

After localization, the tool slices the original report to *only* the statement pages. The extraction prompt assumes a fixed ordering within the slice: slice page 0 is the income statement, slice page 1 is the balance sheet, and slice page 2 is the cash-flow statement if present.

6.4.2 JSON-only extraction prompt

The extraction step uses a strict JSON schema. The prompt requires extracting *all* visible line items and their values by year, preserving signs (parentheses indicate negative values), and using `null` for missing entries.

In the implementation, the Gemini generation configuration explicitly enforces JSON output via `response_mime_type = "application/json"`. This reduces downstream parsing ambiguity and makes the pipeline closer to “tooling” behavior.

6.5 Canonicalization and Accounting Constraints

6.5.1 Canonical fields

Annual reports differ substantially in label conventions (e.g., “cash and cash equivalents” vs. “cash”, “borrowings” vs. “debt”). The script maps extracted labels into a canonical schema using regular expressions.

The canonical balance-sheet fields include

Symbol	Meaning
C	Cash and cash equivalents
AR	Accounts receivable
Inv	Inventories
K	Property, plant and equipment (proxy for fixed assets)
AP	Accounts payable
STD	Short-term debt (incl. current portion of LTD)
LTD	Long-term debt
TA	Total assets
TL	Total liabilities
EQ_{report}	Reported total equity (if present)

The income-statement driver fields include

Symbol	Meaning
REV	Total revenue / net sales
TC	Total costs and expenses
NI	Net income
$EBIT$	Operating income / operating profit
IE	Interest expense (or financial expense proxy)

6.5.2 Hard constraints and diagnostics

For lending analysis, accounting identities must hold. The script writes explicit constraints into `tokens.md` (treated as *hard* constraints for interpretation):

$$E_{\text{imp}} = TA - TL, \tag{6.2}$$

$$TA \geq C + AR + Inv + K, \tag{6.3}$$

$$TL \geq AP + STD + LTD, \tag{6.4}$$

$$C, AR, Inv, K, AP, STD, LTD, TA, TL \geq 0. \tag{6.5}$$

Rather than solving a constrained optimization problem, the current implementation uses these constraints to compute *residual diagnostics*:

$$\text{OtherAssets} = TA - (C + AR + Inv + K), \tag{6.6}$$

$$\text{OtherLiab} = TL - (AP + STD + LTD). \tag{6.7}$$

Large residuals indicate that the canonical mapping failed to classify major categories (e.g., leases, pensions, intangibles) even if the statement is internally consistent at the aggregate level.

6.6 Ratio Computation

All ratios are computed deterministically in Python (to avoid arithmetic errors in the LLM).

6.6.1 Definitions (as implemented)

Let $D = STD + LTD$ denote total debt captured by the canonical mapper, and let equity for leverage ratios be

$$E = \begin{cases} EQ_{\text{report}}, & \text{if reported equity is available,} \\ E_{\text{imp}} = TA - TL, & \text{otherwise.} \end{cases} \tag{6.8}$$

The script computes:

$$\text{Net Income} = NI, \tag{6.9}$$

$$\text{Cost-to-Income} = \frac{TOTAL\_COSTS}{REV}, \tag{6.10}$$

$$\text{Quick Ratio} = \frac{TCA - Inv}{TCL} \quad \text{if } TCA, TCL \text{ exist (with } Inv = 0 \text{ if missing); otherwise } \frac{C + AR}{TCL}, \tag{6.11}$$

$$\text{Debt-to-Equity} = \frac{D}{E}, \tag{6.12}$$

$$\text{Debt-to-Assets} = \frac{D}{TA}, \tag{6.13}$$

$$\text{Debt-to-Capital} = \frac{D}{D + E}, \tag{6.14}$$

$$\text{Debt-to-EBITDA} = \frac{D}{EBIT + DA}, \tag{6.15}$$

$$\text{Interest Coverage} = \frac{EBIT}{|\text{INTEREST\_EXP}|}. \tag{6.16}$$

Here TCA and TCL denote total current assets and total current liabilities if present in the canonical mapping, and DA denotes depreciation/amortization (from cash flow) if extracted.

6.7 Empirical Runs

This section reports results from two annual reports: General Motors (GM) and LVMH. For each run, the pipeline recorded the automatically detected statement pages, extracted multi-year line items, generated canonical tokens, computed ratios, and produced an analyst-style answer.

The generated result are avilable in github, `https://github.com/hrenae/JPMorgan_Internship_Question_1/tree/main/LLMPDF/Gemini/`. In GM and LVMH, we can find corresponding `tokens.md` and `answer.md`.

6.7.1 General Motors (GM) Annual Report

Detected pages. The tool detected income statement at page 60 and balance sheet at page 61 (0-based indexing), and also selected a cash-flow page at 41.[1]

Computed outputs. Table 6.1 summarizes the ratios produced by the automated pipeline for the current year (2023).

Table 6.1: GM (2023) ratios reported by the pipeline.

Metric	Value
Net income	9,840 million
Cost-to-income	0.946
Quick ratio	-0.098
Debt-to-equity	0.0126
Debt-to-assets	0.00313
Debt-to-capital	0.0124
Debt-to-EBITDA	N/A (missing DA)
Interest coverage	10.21×

Accounting diagnostics and failure modes. Although the aggregate balance sheet was internally coherent (reported equity close to implied equity), the leverage ratios are unreliable due to canonical-mapping undercoverage:

- The tokenized debt D was only 856 million, which reflects matching a small “Automotive” debt line while missing the much larger “GM Financial” debt lines. This causes severe understatement of debt-based ratios.
- The computed quick ratio became negative due to a label collision: the regex for TCA can match “Other current assets” rather than “Total current assets”, producing $TCA < Inv$.

[1] The annual report problem statement refers to pages 56–57; offsets can occur due to front matter and PDF indexing differences.

- The liability residual (“OtherLiab”) was extremely large, indicating that most liabilities were not mapped into AP, STD, and LTD categories.

These issues imply that the extracted statements may be arithmetically consistent at the top level while still being unsuitable for credit decisioning without improved canonical mapping.

6.7.2 LVMH Annual Report

Detected pages. For LVMH, the tool detected income statement at page 10, balance sheet at page 25, and cash flow at page 26 (0-based indexing).

Pipeline limitations. The LVMH run illustrates two systematic limitations:

- The canonical revenue (REV) and total cost (TC) fields were missing, preventing automatic cost-to-income computation.
- The regex-based canonical mapper failed to capture borrowings and total liabilities in this report format (e.g., non-breaking hyphen variants in “Short-term borrowings”; and the presence of “Total liabilities and equity” rather than a standalone “Total liabilities” line).

Consequently, the Python-computed metrics were mostly `null` in this run, and a manual override was performed using raw extracted line items to compute core ratios.

Manually computed ratios from raw extract. Using the raw extracted line items, the following values were obtained for the current year (2024): net income = 12,550 (Group share), quick ratio = 0.43, debt-to-equity = 0.33, debt-to-assets = 0.15, debt-to-capital = 0.25, and interest coverage $\approx 23.9\times$; cost-to-income and debt-to-EBITDA were not computable due to missing REV, TC, and DA.

6.8 Generalization to Other Annual Reports

The pipeline is designed to be report-agnostic in the sense that it does not assume a fixed PDF layout, and its extraction step is driven by page localization plus a schema-constrained JSON prompt. Empirically, it can locate and extract statements in both a US GAAP report (GM) and an IFRS-style report (LVMH).

However, the experiments show that the dominant failure mode is **canonical mapping** rather than table extraction itself. To robustly generalize, practical improvements include:

- Use Unicode-normalized regex (to handle non-breaking hyphens and locale-specific punctuation).
- Apply hierarchical matching rules to prefer "total" lines over "other" lines.
- Allow multiple matches (sum/aggregate) for debt components when statements segment debt by business units.
- Expand the canonical schema to include leases, pensions, and intangible-heavy asset classes; otherwise residual diagnostics will remain large for firms like LVMH.

6.9 Reproducibility Notes

The script is parameterized by a Gemini model name passed at runtime. It uses the `google.generativeai` SDK and requires `GOOGLE_API_KEY`. The implementation also sets HTTP/HTTPS proxy environment variables to `127.0.0.1:7890` for network routing.

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] M. Raissi, P. Perdikaris, and G. E. Karniadakis, Journal of Computational physics **378**, 686 (2019).

[6] B. Amos and J. Z. Kolter, in *International Conference on Machine Learning* (PMLR, 2017) pp. 136–145.

[7] B. Lim, S. Arilkumaran, S. Kushwaha, and F. Genthial, International Journal of Forecasting **37**, 1748 (2021).

[8] R. Koenker and G. Bassett Jr, Econometrica: journal of the Econometric Society , 33 (1978).

[9] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, International Journal of Forecasting **36**, 1181 (2020).