Question 1 — Interim Report

Prepared ahead of the November Part 1 submission checkpoint to capture the current state of deterministic and LLM deliverables for Question 1.

1. Executive Summary

- Delivered a deterministic balance-sheet projection stack that enforces accounting identities, augments ratio-based drivers with growth and lagged covariates, and trains a TensorFlow forecaster backed by calibrated bank ensembles for BAC, JPM, and C.
- Processed nine ticker histories via reproducible Yahoo Finance ingestion, validation, and feature pipelines, then evaluated forecasts across assets, equity, earnings, and identity gaps with summarisation/reporting CLIs and pytest coverage.
- Built Part 2 LLM tooling (prompt dataset, adapter, evaluation/comparison CLIs, CFO recommendations) and stress-tested PDF ratio extraction for GM, LVMH, and Tencent; baseline `t5-small` responses lack quantitative coverage and should be combined with deterministic outputs.

2. Literature & Problem Framing

- Vélez-Pareja and Mejía-Pelaez papers guided the identity-preserving projection design; key takeaways are in `reports/q1/literature_summary.md`.
- The deterministic balance-sheet specification covers asset/liability/equity evolution, financing plugs, and simulation framing in `reports/q1/deterministic_balance_sheet_spec.md`.
- Simulation and ML extension roadmaps (exogenous drivers, probabilistic upgrades, backlog) live in `reports/q1/notes/simulation_strategy.md` and `reports/q1/notes/ml_extension_roadmap.md`.

3. Data Acquisition & Processing

- Raw statements for AAPL, MSFT, GM, JPM, BAC, C, HON, CAT, and UNP are cached under `mlcoe_q1/data/raw/*.json` via the `download_statements` CLI.
- Processed balance-sheet/income features come from `prepare_processed_data`, while `build_driver_dataset` constructs ratio/growth/lagged covariates with optional lag filling controls.
- `validate_driver_dataset` checks duplicate periods, missing columns, sparse histories, and filing gaps; pytest covers success and failure paths for data hygiene.

4. Modelling Approach

4.1 Deterministic Projection Layer

- `balance_sheet_constraints.project_forward` enforces Assets = Liabilities + Equity and related tie-outs, reconciling financing gaps after each forecast step.

4.2 Feature Engineering & Forecaster Architecture

- Driver features include leverage, liquidity, profitability, tangible equity, interest spreads, and year-over-year growth, with lag augmentation handled by `augment_with_lagged_features` for temporal context.
- The TensorFlow forecaster combines shared MLP towers with auxiliary sector signals, bank-indicator routing, and complement heads so bank and industrial predictions coexist within a single serialized model.
- Training (`train_forecaster.py`) and evaluation (`evaluate_forecaster.py`) pipelines persist scaling metadata, register custom layers, and emit rich parquet diagnostics for downstream tooling.

4.3 Bank Ensemble Calibration

- Proportional templates (`bank_template.py`) capture liability mix priors per bank, while `BankForecastEnsemble` blends template outputs with neural predictions using calibrated weights saved in `bank_ensemble.json` and auto-applied during evaluation.

5. Forecast Evaluation

Mean absolute errors (billions USD) and identity gaps (billions) averaged over the latest two statement pairs are summarised below; banks use the calibrated ensemble mode.

Ticker	Mode	Assets MAE (B)	Equity MAE (B)	Net Income MAE (B)	Identity Gap (B)
AAPL	mlp	17.74	17.52	67.98	0.000000
BAC	bank_ensemble	<0.01	<0.01	32.17	<0.001
C	bank_ensemble	<0.01	<0.01	14.07	<0.001
CAT	mlp	10.60	14.53	13.20	-0.000000
GM	mlp	18.52	20.44	16.45	0.000000
HON	mlp	10.28	10.58	11.20	NaN
JPM	bank_ensemble	<0.01	<0.01	59.10	<0.001
MSFT	mlp	42.64	46.48	61.47	0.000000
UNP	mlp	55.50	20.28	24.45	0.000000

Bank MAE values are sub-$10 M (hence <0.01 B), equity MAE is sub-$10k, and identity gaps are below $1 M; HON's identity gap is undefined because the source statement omits the necessary liabilities split.

6. PDF Ratio Extraction

- `extract_pdf_ratios` supports GM, LVMH, and Tencent layouts with JSON-configurable strategies, pdfplumber-backed parsing, provenance logging, and pytest-backed heuristics for stable numeric extraction.
- Ratio outputs feed the comparison CLI and documentation for CFO-facing narratives, demonstrating automated computation of net income, cost-to-income, liquidity, leverage, and coverage metrics directly from filings.

7. LLM Benchmarking & Recommendations

- Prompt datasets pair processed statements with ground-truth targets; the HuggingFace adapter runs truncation-aware inference (default `t5-small`) with seeded decoding, while `evaluate_llm_responses` computes coverage, MAE, and MAPE metrics per record.
- Baseline `t5-small` responses produced zero numeric coverage across AAPL and BAC, underscoring the need for stronger models before quantitative reliance. Coverage and error summary:

Ticker	Coverage	MAE (B)	MAPE
AAPL	0.0%	N/A	N/A
BAC	0.0%	N/A	N/A

  Coverage reflects the share of prompts where the model emitted parseable numeric forecasts; MAE/MAPE remain undefined when coverage is zero.

- `compare_llm_and_forecaster` aligns structured and LLM metrics, and `generate_cfo_recommendations` produces Markdown narratives prioritising deterministic forecasts when LLM coverage is low.

8. Testing & Reproducibility

- End-to-end pytest coverage spans driver features, TensorFlow layers, bank ensembles, PDF extraction, LLM adapters, evaluation pipelines, and reporting utilities (`pytest tests/mlcoe_q1 -q`).
- CLIs expose `--help` documentation and produce deterministic parquet/JSON artifacts, enabling reproducible reruns documented in the root README usage examples.

9. Outstanding Opportunities

- Broaden PDF presets to additional issuers and add regression tests for multi-layout coverage.
- Upgrade LLM experiments with higher-coverage models, prompt variants, and quantitative ensembles, tracking robustness across seeds and versions.
- Pursue bonus tracks (credit rating, risk warnings, loan pricing) leveraging existing deterministic and LLM infrastructure as scaffolding.

10. Submission Checklist

- Codebase implemented in Python 3/TensorFlow with TensorFlow Probability ready for probabilistic extensions.
- Comprehensive testing (unit + integration) and documentation assets (`README`, execution plan, literature summaries, status dashboards) committed alongside reproducible data artifacts.