ChatGPT

Synthetic Financial Data Generation for LLM Few-Shot Prompting

Introduction: Why Synthetic Financial Data?

Access to realistic financial data is often limited by privacy, scarcity, or bias issues. Synthetic numerical or tabular business data (e.g. financial statements) offers a workaround by mimicking real data without exposing sensitive information [1] [2]. For tasks like financial analysis, forecasting, or business reasoning with large language models (LLMs), high-quality synthetic data can serve as in-context examples (few-shot prompts) or training augmentation. This helps LLMs learn domain-specific patterns and improves performance, while preserving data privacy and mitigating the need for large labeled datasets [3]. Recent research shows that models trained or prompted with well-crafted synthetic financial data can achieve accuracy on par with those using actual data [4], validating synthetic data as a powerful resource.

LLM-Based Techniques for Synthetic Financial Tables

Direct Prompting with LLMs: Modern LLMs like GPT-4 can be instructed to generate structured financial tables or statements in a zero-shot or few-shot manner. For example, a recent study prompted GPT-4 to produce a synthetic clinical dataset and found the output to be *highly realistic and internally consistent* [5]. The model generated thousands of tabular records (e.g. patient vitals) that correctly preserved logical relations (like Body Mass Index matching height and weight) and closely matched real data distributions on 92% of measured parameters [6]. The authors concluded that LLMs *"can generate realistic tabular synthetic datasets, replicating key statistical properties of real-world data"* [7]. By analogy, prompting an LLM with appropriate instructions (and possibly a few examples) to *"generate a plausible quarterly income statement and balance sheet for a mid-sized tech company"* can yield data that obeys accounting identities and falls in realistic ranges. This approach leverages the LLM's internal knowledge of financial patterns to create synthetic tables with minimal human effort.

Reinforcement and Guided Prompting: More advanced prompting techniques incorporate feedback or structured guidance to improve synthetic data quality. One case is *Reinforcement Prompting* [3], where a smaller policy model iteratively adjusts the prompt given to an LLM ("Executor") in order to produce high-quality financial data. Zuo *et al.* (2024) demonstrate this for financial text data generation: a policy network selects or refines prompts, and GPT-based completion yields synthetic financial statements (e.g. sentiment-labeled finance sentences) [3]. This method preserved data privacy and yielded models trained on the synthetic data that performed *competitively with models trained on real financial data* [4]. The guided approach helps ensure the LLM's outputs meet specific criteria (correct format, diversity, label balance, etc.) via reinforcement learning or prompt optimization.

Few-Shot Prompt Engineering: Another technique is to provide the LLM with a few exemplar tables or statements (possibly themselves synthetic) as a prompt prefix, so the model can mimic the structure and style in its output. Research on *SynAlign* (Ren *et al.*, 2025) highlights the importance of matching the distribution of synthetic data to real data in such few-shot generation. They note that *naively mixing LLM-*

generated text with real data can introduce stylistic and content discrepancies, potentially hurting performance [8]. To combat this, SynAlign has the LLM first summarize key attributes of real examples, then generate new data conditioned on those attributes [9]. After generation, a filtering step (using Maximum Mean Discrepancy) weights or discards synthetic samples to align their overall distribution with the real data [10]. This ensures that the synthetic financial data used in prompts *"closely matches the distribution under which the model will be evaluated"* [11] – a key criterion for high-quality data. In practice, few-shot prompts constructed from such aligned synthetic examples can improve LLM accuracy on financial tasks, as the synthetic examples exhibit realistic tone, content proportions, and variety [8] [12].

Case Study – Synthetic Earnings Statements for Sentiment Analysis: As a concrete example, a Hugging Face project generated ~31k *synthetic financial statement* texts using GPT-based models [13]. These statements (e.g. snippets from earnings reports or market commentary) were labeled positive/negative/neutral and used to augment a small real dataset (~5k samples). The synthetic augmentation *addressed class imbalance and expanded the financial vocabulary* beyond what the real data contained [13]. The result was a sentiment classifier (DistilBERT) that achieved 97.5% accuracy, a significant improvement attributed to the diverse and balanced synthetic data included during training [14]. This underscores that LLMs can be harnessed to generate large quantities of domain-specific financial text which, when used as in-context examples or fine-tuning data, boost model performance on finance tasks.

Programmatic and Simulation-Based Data Generation

Not all synthetic financial data is produced by language models – programmatic and simulation-driven methods are also widely used, especially for numeric tables or transactional data. These approaches rely on algorithms or domain knowledge to create synthetic data that mirrors real-world statistical patterns:

- **Statistical Distribution Modeling:** One industrial example comes from Goldman Sachs, where engineers built a *synthetic data generator for financial contracts* [15]. The generator ingests real contract data (complex JSON structures) and learns the probability distributions of both the structure (paths in the JSON) and the values at each field. It then produces new, fake contracts by sampling from these learned distributions [16]. Crucially, the synthetic contracts maintain *"the same statistical, polymorphic, and structural properties as production data"* and preserve key relational constraints (e.g. keys and nested formats) [15]. The approach involves creating a *"distribution file"* that captures frequencies of various nested fields and value ranges, anonymizing that file, and then generating new samples from it [16]. This yields highly realistic data for internal use (such as testing query pipelines) without exposing any real contract details. The same philosophy can be applied to financial statements: e.g. gather distributions of revenue, expense, and profit margins from real firms, then generate synthetic income statements by sampling from those distributions while enforcing accounting rules (like assets = liabilities+equity).

- **Monte Carlo Simulation & Domain Rules:** In financial forecasting, synthetic time series or statement data can be generated by simulation models. For instance, one might simulate sales growth, costs, and cash flows under random market conditions to produce many plausible financial projections for a company. Tools like PaySim (for payments data) use agent-based simulations calibrated to real transaction patterns to create synthetic datasets for fraud detection research [17]. The advantage of simulation is fine-grained control: developers can inject specific scenarios (e.g. a recession year, or a one-time impairment expense) to ensure the synthetic data covers diverse situations that an LLM might need to reason about. Programmatic rule-based generation also

guarantees logical consistency. For example, one can generate a balance sheet by first randomizing asset categories, then computing liabilities and equity to *exactly balance*, thus always respecting fundamental accounting equations.

- **Generative Models (VAEs/GANs):** Outside of LLMs, other AI models like Variational Autoencoders and Generative Adversarial Networks have been applied to financial data. **Motai *et al.* (2025)** show that a VAE can learn from confidential accounting records and output *synthetic journal entries* that closely match the statistical properties of the real entries [2] [18]. Their synthetic data preserved the *double-entry bookkeeping structure* (debits and credits) and was evaluated quantitatively to ensure it was an accurate proxy for the original data [19] [20]. This demonstrates a way to generate realistic numerical datasets (e.g. transaction ledgers or financial ratios) using neural generative models rather than hand-crafted rules. Similarly, researchers have proposed transformer-based generative models for tabular data (sometimes called TabGPT or TTVAE) that can handle mixed numeric and categorical features [21] [22]. Such models could, in theory, be trained on public financial data and then sampled to produce unlimited synthetic financial statements that retain complex correlations present in real companies' data.

Each programmatic approach has trade-offs. Rule-based simulation offers interpretability and guaranteed validity (no nonsensical outputs), but may require expert tuning to mimic real-world variability. Deep generative models can automatically capture correlations (e.g. between revenue and expenses) but risk overfitting or reproducing traces of the training data if not carefully regularized. In practice, a hybrid strategy is often effective: use real data to inform distributions or train a model, then inject randomness and privacy safeguards to ensure the output is both realistic and novel.

Quality and Effectiveness of Synthetic Data for LLMs

Improving Model Performance: A primary motivator for using synthetic data in few-shot prompts is to boost model performance on tasks where real examples are limited. Multiple studies report that carefully generated synthetic datasets can *significantly improve accuracy and generalization*. In the SynAlign work, augmenting a weak retriever system with LLM-synthesized text (filtered for distribution match) led to *"significant performance improvements"* on downstream language tasks [12]. Likewise, Zuo et al. found their LLM-generated financial text data allowed sentiment classifiers to **bridge the performance gap** between training on scarce real data vs. abundant synthetic data [23]. In few-shot prompting scenarios, including relevant synthetic examples can help an LLM follow the correct format or reasoning steps. For example, an LLM asked to perform a financial ratio analysis might do better if the prompt first shows a **fabricated** mini financial statement and an example analysis of it. The synthetic example sets the context, and because it's fabricated, one ensures the model isn't relying on memorized real company facts but truly applying reasoning. Research by Peng *et al.* (2023) on stance detection even demonstrated that *LLM-generated synthetic data can surpass human-written data in some cases*, when used to fine-tune models [8] [12] – the key is that the synthetic data must be diverse and task-aligned.

Fidelity and Realism Evaluations: Ensuring synthetic data is *realistic enough* is crucial. Studies have introduced various metrics to evaluate fidelity. In the GPT-4 clinical data study, statistical tests (t-tests, proportion tests) were used to compare synthetic vs. real data distributions on each feature [24] [6]. High p-values and overlapping confidence intervals indicated that the LLM's synthetic data was virtually indistinguishable from real data on most fields [6]. In financial contexts, researchers might compare summary statistics (means, variances, correlations) of synthetic statements to those of real companies. The

VAE-generated journal entries mentioned above were *"quantitatively evaluated for quality"* to ensure they maintained realistic patterns [19] [20]. Another simple check is internal consistency: e.g., does a generated balance sheet actually balance to the penny? Does a synthetic cash flow statement's ending cash equal the balance sheet's cash change? LLMs guided with chain-of-thought prompting can be made to self-verify such constraints. Indeed, GPT-4 has shown the ability to respect arithmetic relations in output when prompted carefully, as evidenced by the BMI calculation example in the clinical study (no inconsistencies in 6,166 cases) [5]. These evaluations build confidence that synthetic data will not mislead the model during few-shot learning – in other words, the LLM treating the synthetic example as "ground truth" is acceptable because it statistically resembles ground truth.

Effect on Few-Shot Prompting: In practice, users have found that a few well-chosen synthetic examples can **guide LLM reasoning** better than zero-shot prompts. The few-shot prompts provide a template for the answer structure and highlight relevant details. If the synthetic examples are too simplistic or unrealistic, the LLM might produce superficial answers; but if they are richly patterned after real data, the LLM picks up those nuances. For instance, giving GPT-3 a synthetic financial report with subtle cues (like seasonality in revenue or an unusual one-time loss) can prompt it to discuss those nuances in its analysis of a new case. The synthetic data essentially functions as *task-specific training on the fly*. However, as cautioned by Ren *et al.*, one must avoid synthetic examples that are out-of-distribution – e.g., overly idealized or erroneous data – since that can distort the model's expectations [25]. When synthetic data matches the diversity and style of real data, few-shot prompting can yield outputs that analysts find comparable to those using real historical cases.

Ensuring Realism, Diversity, and Avoiding Memorization

When generating synthetic financial data, several best practices emerge from the literature:

- **Maintain Realism with Constraints:** Ensuring that synthetic data obey domain rules improves realism. This can be achieved by hard constraints in programmatic generation (as with balance sheet equations or journal-entry double entries) or by instructing LLMs to follow specific formats/calculations. For example, an LLM prompt might include: *"Note: Ensure that the balance sheet balances and all ratios are plausible."* The Frontiers study showed that even zero-shot GPT-4 could respect logical constraints when asked (e.g. it correctly computed derived fields like BMI) [5]. Similarly, Goldman's generator preserved complex JSON schema rules by design [26] [27]. The takeaway is to bake domain knowledge into the generation process so that synthetic data *"looks and feels"* legitimate to a business analyst or an AI model.

- **Maximize Diversity:** Synthetic data should cover a wide range of scenarios to be most useful. LLMs have a tendency to produce *average* cases if not prompted for variety. Techniques like *exploration-aware prompt selection* (in SynAlign) explicitly push the generation into less-covered regions of the data space [28]. In practical terms, one might prompt an LLM with different industry contexts, outlier conditions, or rare events to ensure the few-shot examples are not all alike. The Hugging Face project specifically cited *"diverse financial vocabulary coverage"* as a benefit of LLM-generated statements [29] – the synthetic data introduced new phrases and sentiments that enrich the model's understanding. One can also randomize numerical aspects: e.g., vary company sizes, years, or growth rates in synthetic financial statements, so the model doesn't overly tune itself to one pattern. Diversity reduces the chance of an LLM overfitting to idiosyncrasies of a narrow synthetic dataset.

- **Avoiding Training Data Leakage:** A subtle concern is that an LLM might regurgitate parts of its training data when asked to generate financial data, especially if the prompt inadvertently cues a well-known example. To avoid *memorized patterns*, researchers recommend steering models toward fictitious entities and figures. For example, instead of prompting with *"Generate Apple Inc.'s balance sheet for 2022"* (which the model might have seen), one could ask for *"a hypothetical tech firm's balance sheet"*. Reinforcement prompting approaches inherently favor novel outputs because the policy model can be optimized to penalize copies. Additionally, some works suggest filtering out synthetic outputs that are too similar to any known real sample (using similarity search) as a post-processing step. In the context of few-shot prompting, the goal is to ensure the LLM is learning from the *concepts* in the synthetic data (e.g. how a recession impacts financials) rather than memorizing specific real-world values. Using anonymized distributions (as in the GS contract example) or broad statistical templates helps in this regard [26] [30]. Indeed, Goldman's process explicitly anonymizes and then regenerates data, breaking any direct linkage to real entities [31].

- **Evaluation and Iteration:** Finally, maintaining realism and usefulness is an iterative process. After generating synthetic data, it should be tested either by human domain experts or by the models themselves. For instance, one might run a trained model on synthetic inputs to see if any obviously wrong conclusions are reached – if so, the synthetic data may contain implausible quirks that need fixing. Quantitative checks (distribution alignment tests, etc.) can flag if synthetic data has drifted from reality on key metrics (e.g., average profit margins far from any real company's). Many studies emphasize *filtering* synthetic data: not every LLM output is kept blindly [10]. By culling low-quality or repetitive samples and focusing on those that add new information, the final synthetic dataset remains both **diverse and high-fidelity**.

Conclusion and Best Practices

Synthetic financial data generation is a rapidly evolving field, leveraging both powerful LLMs and traditional simulation techniques to create realistic business data for AI applications. Key takeaways from research and case studies include:

- **Leverage LLM Knowledge:** Large models can generate rich financial statements or tables when guided with the right prompts or reinforcement loops. They encapsulate domain patterns (linguistic and numerical) learned from vast training data [3], making them useful "engines" for synthetic data creation, especially when expert crafting of prompts or multi-step generation is applied.

- **Use Domain-Driven Simulation:** In tandem, programmatic approaches (statistical modeling, Monte Carlo, VAEs/GANs) ensure that numeric relationships and distributions remain realistic [19] [15]. These approaches can fill gaps by producing data in areas where the LLM might be less reliable (for example, extremely high-dimensional data or strictly formatted records).

- **Validate and Align Data:** Always compare synthetic data against real data (when available) to validate its quality. Techniques like distribution alignment (SynAlign's MMD weighting) and constraint checking are valuable to keep synthetic outputs on-track [9] [6]. High-quality synthetic data should be statistically indistinguishable from real data in the aspects that matter for the task.

- **Enhance Few-Shot Prompts with Variety:** When using synthetic data in prompts, include a breadth of scenarios to maximize the prompt's informativeness. Because you control the generation, you can

ensure that each example in a few-shot prompt highlights a different facet of the task (e.g. one prompt example shows a profitable year, another shows a loss, etc.). This helps the LLM generalize better in its responses.

- **Monitor for Bias and Memorization:** Synthetic does not automatically mean unbiased or original – an LLM might carry biases from training data into its synthetic outputs, or a simulation might inadvertently ignore minority cases. Be mindful of these and, if needed, adjust the generation process (for instance, prompt the LLM to include diverse industry sectors, or augment a simulator with rare events). Also verify that no sensitive or proprietary information leaked into the synthetic data [1].

By following these practices, practitioners have successfully used synthetic financial data to improve LLM-driven analysis and decision support. As one study put it, *"high-quality synthetic data…closely matching the evaluation distribution"* is often *"the optimal dataset"* for model training or prompting [11]. In financial applications where real data is scarce or sensitive, synthetic data generation—whether via LLMs, algorithms, or hybrids—opens the door for robust few-shot learning, better forecasting, and more powerful business reasoning with AI.

Sources: The insights above are drawn from recent research on LLM-generated synthetic datasets [5] [4], industry case studies of programmatic data simulators [15] [16], and evaluations of synthetic data effectiveness in finance-related AI tasks [13] [19]. These studies collectively highlight the techniques and considerations crucial for generating synthetic financial statement data that is realistic, diverse, and beneficial for large language models.

---

[1] [3] [4] [23] Reinforcement prompting for financial synthetic data generation · 研飞ivySCI
https://www.ivysci.com/en/articles/4098271__Reinforcement_prompting_for_financial_synthetic_data_generation

[2] [18] [19] [20] Generating Synthetic Journal-Entry Data Using Variational Autoencoder - Peeref
https://www.peeref.com/zh/works/85810772

[5] [6] [7] [24] Frontiers | Large language models generating synthetic clinical datasets: a feasibility and comparative analysis with real-world perioperative data
https://www.frontiersin.org/journals/artificial-intelligence/articles/10.3389/frai.2025.1533508/full

[8] [9] [10] [11] [12] [25] [28] Few-shot LLM Synthetic Data with Distribution Matching
https://arxiv.org/html/2502.08661v1

[13] [14] [29] AdityaAI9/distilbert_finance_sentiment_analysis · Hugging Face
https://huggingface.co/AdityaAI9/distilbert_finance_sentiment_analysis

[15] [16] [26] [27] [30] [31] Synthetic Data Generator for Financial Contracts - Goldman Sachs Developer
https://developer.gs.com/blog/posts/synthetic-data-generator

[17] Synthetic Financial Datasets For Fraud Detection - Kaggle
https://www.kaggle.com/datasets/ealaxi/paysim1

[21] What is synthetic data? - IBM Research
https://research.ibm.com/blog/what-is-synthetic-data

22 TTVAE: Transformer-based generative modeling for tabular data ...
https://www.sciencedirect.com/science/article/pii/S0004370225000116