Decision Transformer for Crypto Trading in FinRL

Trajectories → Training → Backtest with Costs

BTC-USD backtest (~35k steps): Sharpe 0.27 vs. 0.11 benchmark; Return −4.6% (MDD −9.5%, Vol 55%)

Authors: Alexei Savin

Goal & Pipeline

Goal. Evaluate a Decision Transformer (DT) for BTC-USD trading within the FinRL pipeline under realistic costs. DT treats reinforcement learning as sequence modeling: conditioned on return-to-go (RTG), past states, and past actions, it predicts the next action. This enables offline training and mitigates instability from non-stationary markets.

Pipeline.

1. **Data:** load OHLCV for selected pairs and engineer indicators.
2. **Environment:** FinRL gym-style crypto market with transaction costs and optional risk constraints.
3. **Trajectories:** create (state, action, reward) sequences from historical strategies or baseline RL.
4. **Train DT** on trajectories with RTG conditioning.
5. **Validation & backtest** with commission and slippage.
6. **Reporting:** equity curve, risk metrics, and ablations.

Data, Environment, Method

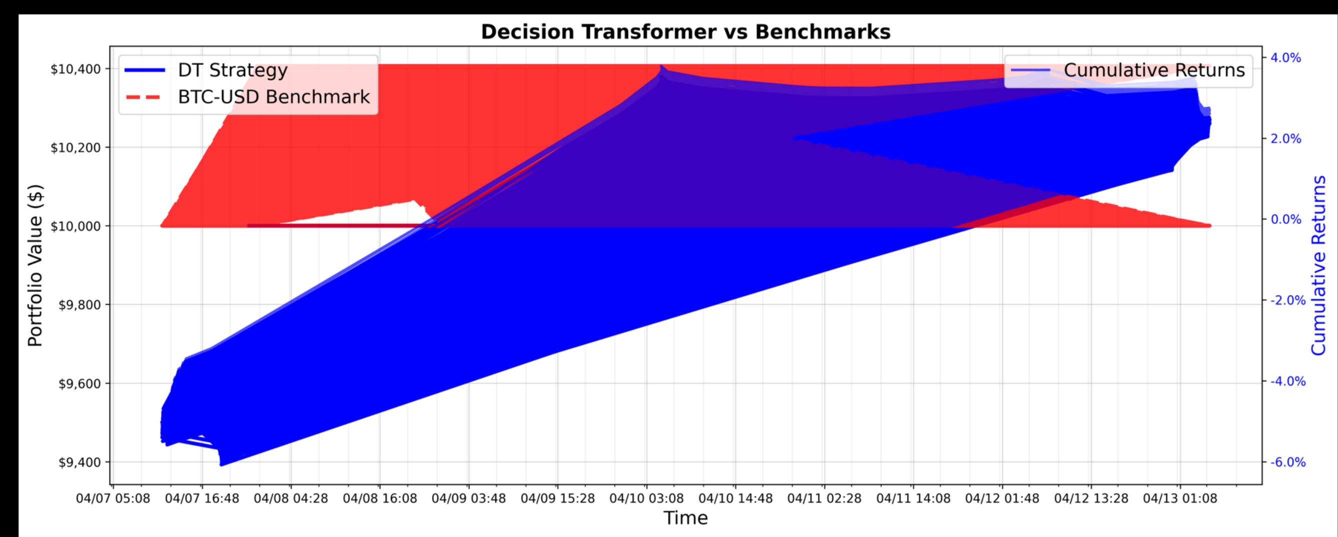

Fig. 1. Equity curve (net of costs): DT vs. BTC-USD benchmark on the test span.

Data. Single BTC-USD test span with ~35,000 aligned steps. Features include prices, volume, returns, rolling volatility, and standard technical indicators (EMA, RSI, MACD). Splits are chronological to avoid leakage.

Environment. Observations combine market features with position and cash. Actions represent position changes (e.g., long/flat/short or target weight). Reward equals portfolio value change **net of** commission and slippage. Optional constraints limit exposure and drawdown.

Method (Decision Transformer). Inputs are tokenized as $[\mathrm{RTG},\ state_t,\ action_{\{t-1\}}]$ across a context window K. A decoder-only Transformer outputs the next action. We normalize RTG, clip rewards, and apply early stopping. For discrete actions we use cross-entropy; for continuous weights we use MSE with action squashing. Ablations vary K, feature sets, and cost assumptions.

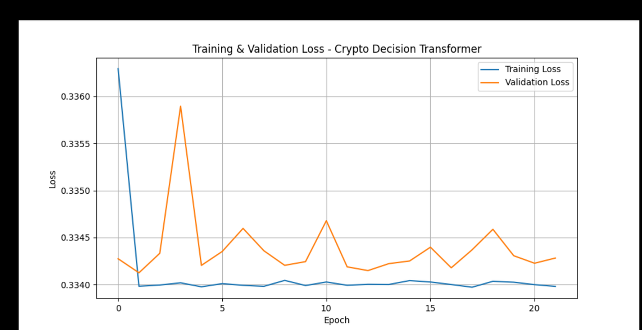

Fig. 2 Training and validation loss.

Results, Insights, Next Steps

Results (test, net of costs). DT achieves **Sharpe 0.27** vs **0.11** for BTC-USD Buy-and-Hold, with **Total Return −4.6%**, **Max Drawdown −9.5%**, and **Annual Volatility 55%**. Benchmark achieves **Return 4.1%**, **MaxDD −3.9%**, **Vol 22%**. Risk-adjusted gain improves (Sharpe), while return lags in this short evaluation.

Trading Statistics. Total actions: **35,000**; Buys **7,137 (20.4%)**; Sells **0**; Holds **27,863 (79.6%)**; Total trades **7,137**.

Prediction vs Execution. Predictions: Buy **34,999 (100%)**, Hold 0, Sell 0. Executed: Buy **7,137**, Hold **27,863**, Sell 0. **Mismatch: 79.6%** of steps (prediction ≠ execution).

Next Steps. Refine action discretization and execution rules; add PPO baseline and equity curves; run context-length K ablation; enrich features (order-book/on-chain); explore cost-aware augmentation and DT+actor-critic ensembles.

Model	Total Return	Annualized	Sharpe	Max Drawdown	Annual Volatility	Calmar	Win-rate	Final Value
DT (ours)	**−4.61%**	−0.03%	**0.274**	**−9.50%**	**55.08%**	−0.003	**49.96%**	**$9,539.49**
BTC-USD (BH)	4.06%	0.03%	0.112	−3.90%	22.16%	0.007	6.15%	$10,405.90

Table 1. Performance metrics after costs (Return, Sharpe, MaxDD, Volatility).

Sharpe: 0.27 (DT) vs. 0.11 (BH)

MaxDD: −9.5% (DT) vs. −3.9% (BH)

Volatility: 55% (DT) vs. 22% (BH)

Return: −4.6% (DT) vs. +4.1% (BH)

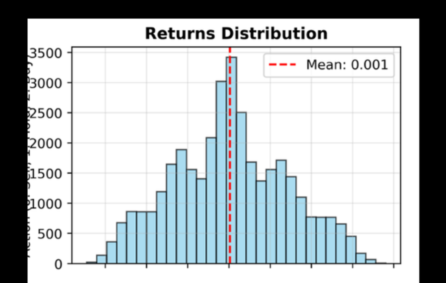

Fig. 3 Distribution of step returns on the test set.

ABOUT/CONTACTS

GitHub

Telegram

LinkedIn

FinGPT-Powered Compliance Agents for SecureFinAI 2025

Multi-task evaluation of a LoRA-augmented financial LLM on Financial Q&A, Sentiment, and XBRL extraction

Task_2_FinGPT_Powered_Compliance_Agents

Authors: Alexei Savin

MLX

Objective & Problem

Objective.

Design and evaluate an LLM-based compliance agent that can (1) answer finance/compliance questions, (2) classify the sentiment of financial text, and (3) extract structured fields from XBRL-like reports. The agent should run with a lightweight **PEFT / LoRA** adapter and produce **auditable, source-grounded outputs**.

Motivation.

Compliance and financial supervision tasks are document-heavy and multi-format: short Q&A on rules, long disclosures with sentiment, and structured filings (XBRL). A single agent that can route the user request to the right subtask and return a scored result is more useful than separate scripts.

Agent capabilities.

1. Financial Q&A - answer domain-specific questions from an evaluation set.
2. Sentiment analysis - classify financial phrases and news into positive / neutral / negative
3. XBRL extraction - read structured samples and recover tags, values, and formulae.

Runtime setup.

The repository is evaluated via `make evaluate`. All metrics are saved to `results/evaluation_results.json` for reproducibility.

Architecture & Workflow

- **Model & configuration.**
  - **Mode:** **peft**
  - **Adapter ID:** xsa-dev/fingpt-compliance-agents
  - PEFT base model: meta-llama/Llama-3.2-1B-Instruct
  - Baseline model: meta-llama/Llama-3.2-1B-Instruct
  - Baseline Device: **mps**
  - PEFT Device: Cuda **1x RTX 4090**
- Workflow.
- 1. Load the FinGPT-style financial LLM and attach the LoRA adapter.
- 2. Select task based on the evaluation pipeline: Financial Q&A → Sentiment → XBRL.
- 3. Run task-specific evaluator (accuracy, F1, precision/recall, or XBRL field-level metrics).
- 4. Aggregate results - into an overall score.
- Why multi-task?
- Compliance scenarios rarely consist of only one kind of question. A realistic system must:
  - read a rule and answer a question about it (Q&A),
  - detect tone/polarity of an event or disclosure (sentiment),
  - and pull exact numbers/tags from filings (XBRL).
- Evaluating on all three makes the agent closer to a real “compliance assistant.”
- Desired output format.
- Verdict - Compliant / Partially compliant / Non-compliant (or: Answer found / Not found)
- Evidence - source file / section / tag used to answer
- Reasoning - 3–5 sentences grounded on retrieved content

- Baseline Results (this run)

- Financial Q&A — Accuracy 0.677 (n=31)
- Sentiment (baseline evaluator) — Acc 0.418, F1 0.453, Prec 0.546, Rec 0.418 (n=826)
- XBRL extraction — Overall accuracy 0.920 (Tag 0.915, Value 0.909, Formula-construct 0.982, Formula-calc 0.885) (n=574)
- Overall (aggregated) — Accuracy 0.548, F1 0.453, Precision 0.546, Recall 0.418

Task	Metric(s)	Score	n
Financial Q&A	Accuracy	0.677	31
Sentiment (baseline)	Acc / F1 / Prec / Rec	0.418 / 0.453 / 0.546 / 0.418	826
XBRL extraction	Overall XBRL accuracy	0.920	574
Overall	Accuracy / F1	0.548 / 0.453	—

PEFT Sentiment (separate eval on en-fpb)

Config: PEFT (LoRA) on meta-llama/Llama-3.2-1B-Instruct, dataset ChanceFocus/en-fpb (test n=970)
Accuracy: 0.618

Class-wise (PEFT): Neutral 97.4% | Positive 7.9% | Negative 12.9%
Takeaway: domain-specific PEFT lifts sentiment from ~0.42 → ~0.62,
but positives/negatives remain under-detected (class imbalance).

Conclusions & Next Steps

Strongest: XBRL (~0.92). Financial Q&A: acceptable (~0.68). Weakest: sentiment (~0.42 baseline). PEFT improves sentiment to ~0.62 on a finance dataset; focus on class rebalancing and prompts. Add retrieval/grounding for Q&A; map XBRL fields to checklists (field → rule → pass/fail); expose as auditable JSON report.

ABOUT/CONTACTS

GitHub

Telegram

LinkedIn

HuggingFace

Planned work: extension to multi-modal FinGPT agents (Task 3) — prototype only, not reported here.