WaveGAN: Wavelet-Augmented Conditional GAN for Short-Horizon BTCUSDT Extrapolation

Jashua Luna

September 10, 2022

Abstract

This report presents WaveGAN, a wavelet-augmented conditional generative adversarial framework for short-horizon extrapolation of BTCUSDT minute data. Rather than forecasting directly in the time domain, the method transforms each price-derived signal into a structured wavelet-history tensor and learns a conditional mapping from an observed tensor to a future tensor. The implementation evaluates three prediction lengths, 30, 60, and 128 minutes, using paired wavelet tensors generated from BTCUSDT minute shards. Across all three horizons, the trained model improves substantially over an identity-style baseline in the wavelet domain, with mean holdout RMSE values of 0.1885, 0.2432, and 0.2799, respectively. The results show that the WaveGAN formulation captures meaningful transformed-domain structure while also revealing the increasing difficulty of longer-horizon extrapolation.

1 Introduction

Short-horizon financial extrapolation is difficult because the signal of interest is noisy and changes across multiple time scales. WaveGAN addresses this problem by shifting the prediction task into the wavelet domain. Instead of asking a model to predict a future price-derived sequence directly, the method constructs a history of dual-tree complex wavelet coefficients and treats the forecasting problem as a conditional image-to-image translation task.

In WaveGAN, each training example consists of an input tensor representing recent wavelet-history structure and an output tensor representing the future wavelet-history structure to be extrapolated. A conditional generator maps the input tensor to a predicted future tensor, while a discriminator encourages the generated tensor to remain consistent with the statistics of the target domain. This design makes it possible to model local and coarse structure jointly across wavelet levels.

The experiments reported here focus on BTCUSDT minute data and evaluate three prediction lengths: 30, 60, and 128 minutes. The purpose of the study is to determine whether the WaveGAN formulation can learn a useful transformed-domain mapping and how that ability changes as the forecasting horizon increases.

2 Model Background

WaveGAN combines wavelet-domain signal representation with conditional adversarial learning. The generator-discriminator design follows the image-to-image translation paradigm introduced by pix2pix, particularly the use of a U-Net style encoder-decoder generator with skip connections and a PatchGAN-style discriminator [1]. The MATLAB implementation also draws on established public MATLAB GAN examples [2]. Minute-level BTCUSDT data acquisition is supported through the Python helper `Get_Data.py`, which relies on the `python-binance` ecosystem [3]. The transform stage is built around MATLAB's dual-tree complex wavelet functionality and should be interpreted in the context of the standard dual-tree complex wavelet transform literature [4].

The key design choice in WaveGAN is to represent each sample as a wavelet-history tensor rather than as a raw time series. This tensor interleaves real and imaginary coefficient channels and preserves structure across wavelet levels, allowing the conditional GAN to operate on a richer description of recent and future dynamics than a one-dimensional sequence alone would provide.

3 Experimental Setup

3.1 Data and Configuration

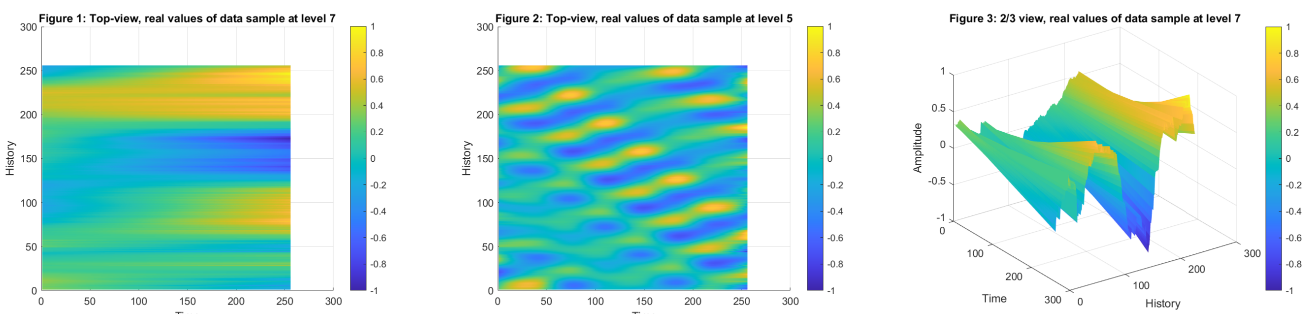
The study uses BTCUSDT minute-level data partitioned into weekly text shards. Those raw shards are stored under `data/raw/binanceData`. During preprocessing, the implementation draws fixed-length windows from the raw data, converts them into paired input-output wavelet-history tensors, and stores the resulting MATLAB datasets under `data/processed`. The default experiment configuration is defined in `src/+wavegan/defaultConfig.m`.

The experiments in this report use a window size of 256 minutes and evaluate three prediction lengths: 30, 60, and 128 minutes. For each horizon, the data-generation stage constructs 1000 paired samples. Training is run for up to 200 epochs with Adam-based optimization, a batch size of 1, and a holdout fraction of 20%. The implementation requires MATLAB with Deep Learning Toolbox and Wavelet Toolbox. Parallel Computing Toolbox is helpful for data generation and compatibility execution paths, and a CUDA-capable GPU is useful for efficient training.

3.2 Wavelet-History Representation

WaveGAN models transformed-domain structure rather than direct price trajectories. For each sample, the input signal is processed repeatedly through the complex wavelet transform over a moving window. The resulting histories of wavelet coefficients are interpolated to a fixed tensor shape and arranged into a multi-channel representation. The generator then predicts the future tensor conditioned on the observed tensor.

Figure 1 shows representative views of an input tensor from the 30-minute experiment. These plots illustrate the kind of structure presented to the model: a layered description of wavelet coefficients whose geometry depends on both scale and temporal context.



(a) Level 7, top view. (b) Level 5, top view. (c) Level 7, oblique view.

Figure 1: Representative views of the WaveGAN input tensor for the 30-minute horizon. The surfaces show real-valued wavelet-history slices at two levels and illustrate the structured tensor presented to the generator.

4 Validation Protocol

Evaluation is performed in the wavelet domain because the implementation focuses on the conditional generator that predicts future wavelet-history tensors. Each run saves a checkpoint, holds out a validation subset, evaluates the generator on that holdout set, and generates report-oriented figures from the saved artifacts. Summary rows are written to `outputs/experiment_summary.csv`, checkpoint and evaluation artifacts are written under `outputs/checkpoints`, and figure assets are copied under `reports/figures`.

With 1000 generated samples per horizon and a holdout fraction of 20%, each experiment evaluates on 200 held-out samples. The primary metric is wavelet-domain root mean squared error:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\frac{1}{WHC}\sum_{x=1}^{W}\sum_{y=1}^{H}\sum_{c=1}^{C}\left(Y_{x,y,c}^{(n)} - \hat{Y}_{x,y,c}^{(n)}\right)^2}.$$

Here W, H, and C denote tensor width, height, and channel count, and $\hat{Y}$ is the generator output. A simple identity-style baseline is also reported by treating the input tensor as though it were the future tensor and measuring its RMSE against the true output. This baseline provides a direct test of whether the model is learning a meaningful conditional mapping rather than merely copying or preserving observed structure.

Training uses early stopping based on validation RMSE, and the best validation checkpoint is restored before final evaluation. As a result, the reported holdout metrics correspond to the best saved model state for each horizon.

5 Results

5.1 Quantitative Summary

WaveGAN shows a clear monotonic increase in difficulty as the prediction horizon grows. Table 1 reports mean holdout RMSE values of 0.1885 for the 30-minute horizon, 0.2432 for the 60-minute horizon, and 0.2799 for the 128-minute horizon. The same ordering appears in the median RMSE values and is accompanied by modest but consistent variation across held-out samples.

Horizon	Mean RMSE	Median RMSE	Std. RMSE	Baseline RMSE	Improvement
30	0.1885	0.1864	0.0217	0.3828	50.76%
60	0.2432	0.2436	0.0210	0.3916	37.91%
128	0.2799	0.2827	0.0228	0.3722	24.79%

Table 1: Holdout wavelet-domain RMSE summary for the three prediction horizons. Improvement is computed relative to the identity-style baseline.

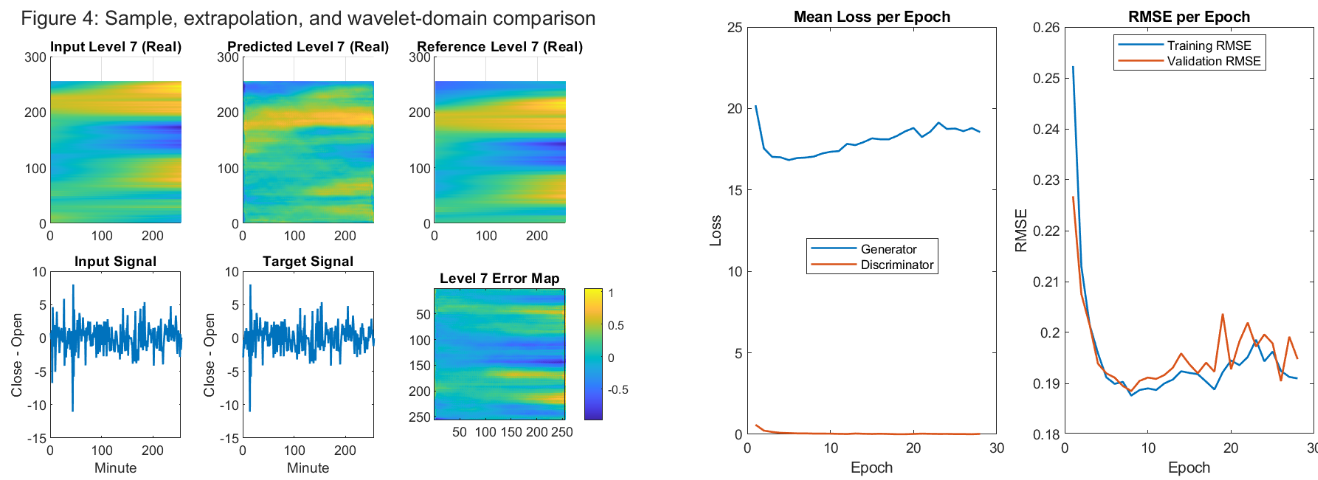
All three models outperform the identity baseline by a substantial margin. Relative to baseline mean RMSE, the trained generator reduces error by 50.76% at 30 minutes, 37.91% at 60 minutes, and 24.79% at 128 minutes. These gains show that the model is learning future-oriented structure in the transformed domain rather than simply reproducing the observed tensor.

The training records also reveal a consistent pattern across horizons. The 30-minute experiment reaches its best validation RMSE at epoch 8 and stops after 28 epochs. The 60-minute experiment reaches its best value at epoch 12 and stops after 32 epochs. The 128-minute experiment is more fragile: it stops after 21 epochs, and its best validation value occurs at the earliest checkpoint. This progression matches the numerical summary and suggests that optimization becomes less forgiving as the extrapolation problem becomes longer range.

5.2 30-Minute Horizon

The 30-minute experiment is the strongest overall result in the study. It has the lowest holdout RMSE, the largest gain over the identity baseline, and the most stable training behavior. In Figure 2, the predicted level-7 structure remains visually close to the target structure, and the error map is comparatively localized.

The diagnostics panel shows a sharp early drop in validation RMSE followed by a stable plateau near the best recorded value.



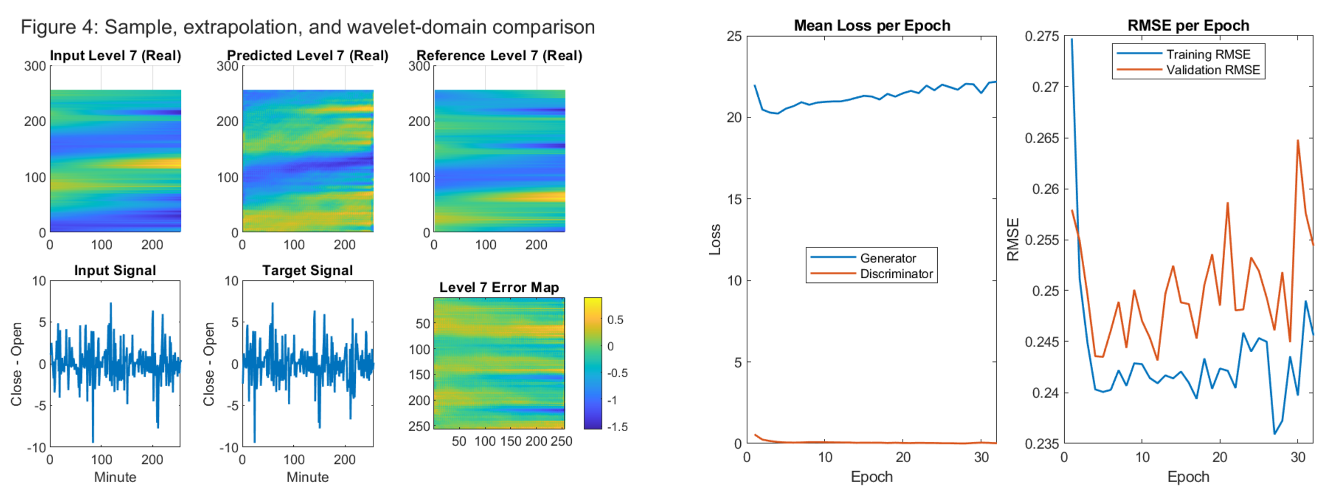
(a) Sample-level comparison.

(b) Loss and RMSE diagnostics.

Figure 2: WaveGAN results for the 30-minute horizon. The prediction remains close to the reference structure, and the training curves show the most favorable behavior in the suite.

5.3 60-Minute Horizon

The 60-minute experiment remains clearly better than the identity baseline, but the forecasting task is visibly harder. Holdout RMSE increases, the relative gain over baseline decreases, and the validation curve settles into a broader plateau. Even so, Figure 3 still shows a structured prediction rather than a collapse to noise or a trivial copy of the input, indicating that the WaveGAN formulation retains useful predictive signal at this intermediate horizon.



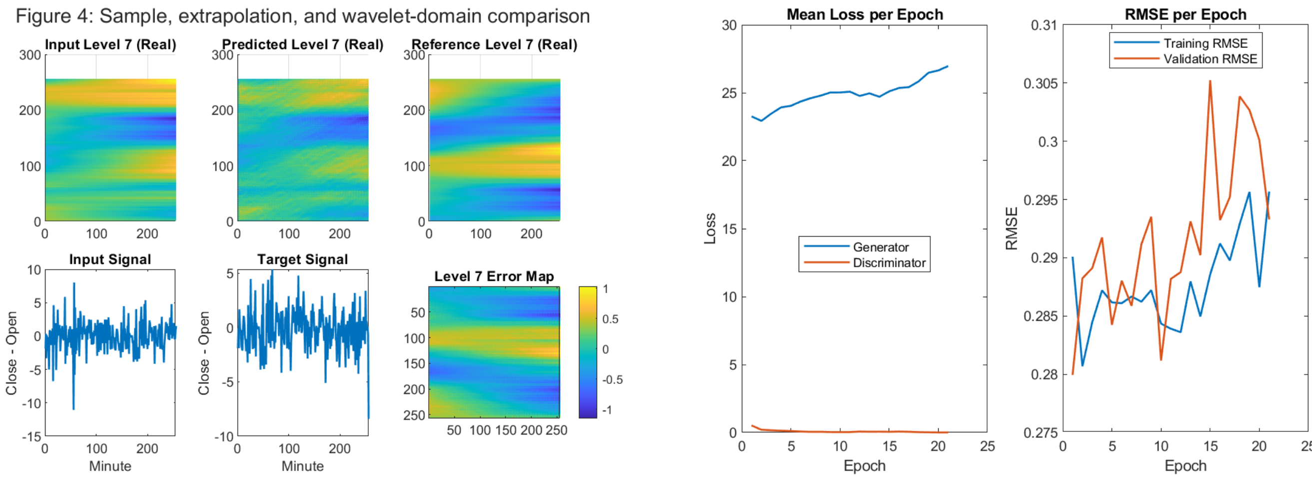
(a) Sample-level comparison.

(b) Loss and RMSE diagnostics.

Figure 3: WaveGAN results for the 60-minute horizon. The model continues to capture meaningful transformed-domain structure, but with broader errors than in the 30-minute case.

5.4 128-Minute Horizon

The 128-minute experiment is the hardest case considered here. The model still improves over the identity baseline, which means it has not lost predictive value altogether, but the gain is smaller and the optimization path is noticeably less stable. Figure 4 shows a rougher prediction-error pattern, and the diagnostics indicate that validation performance does not improve beyond the earliest saved checkpoint. The longest-horizon result should therefore be interpreted as partial transformed-domain signal capture rather than a high-fidelity extrapolation.



(a) Sample-level comparison.

(b) Loss and RMSE diagnostics.

Figure 4: WaveGAN results for the 128-minute horizon. The model still beats the identity baseline, but the prediction is less accurate and the optimization behavior is less stable than at shorter horizons.

6 Discussion

The experiments show that WaveGAN learns a non-trivial conditional mapping in the wavelet domain. Across all three horizons, the trained model improves substantially over the identity baseline, and the qualitative figures align with the numerical error summaries. The strongest performance appears at the 30-minute horizon, where both the comparison plots and the training diagnostics suggest that the model is capturing large-scale transformed-domain structure in a stable way.

At the same time, the results make the limits of the method clear. Performance degrades steadily with prediction length, and the 128-minute experiment is significantly less stable than the shorter-horizon runs. This indicates that the wavelet-history representation is informative, but it does not remove the underlying difficulty of forecasting farther ahead in a highly variable financial signal.

The current implementation should therefore be interpreted as a transformed-domain forecasting system rather than a complete end-to-end market prediction pipeline. Its main empirical claim is that recent wavelet-history structure contains enough information for a conditional GAN to produce improved future tensors over a naive carry-forward baseline. That claim is well supported by the present experiments.

7 Future Work

Several extensions would strengthen the study. The first is to repeat the experiments across multiple random seeds and report uncertainty intervals rather than single-run point estimates. The second is to compare WaveGAN against stronger baselines, including direct convolutional regressors and classical forecasting methods. The third is to perform focused ablations on scaling, holdout construction, tensor design, and horizon length to isolate which parts of the representation contribute most to performance.

The most important methodological extension is to add an inverse-transform stage that maps the predicted wavelet-domain tensor back into a directly evaluated time-domain forecast. Without that stage, the present implementation demonstrates transformed-domain extrapolation quality but does not yet establish time-domain forecasting quality in the source signal space.

A Experiment Pipeline

The full experiment suite is launched from `experiments/run_report_suite.m`. A complete execution performs the following steps:

1. Load the default configuration and construct experiment runs for prediction lengths 30, 60, and 128 minutes.
2. Check for the processed MATLAB datasets in `data/processed`; if they are missing, generate them from the BTCUSDT shards in `data/raw/binanceData`.
3. Sample raw price-derived sequences, compute dual-tree complex wavelet transforms over moving windows, interpolate the transform histories to a fixed tensor shape, and interleave real and imaginary channels to form paired input-output tensors.
4. Save each generated dataset as a MATLAB file containing `inputData`, `outputData`, and supporting signal-level bookkeeping.
5. Train the first-stage conditional GAN for each horizon using the grouped-convolution architecture, Adam-based optimization, and the combined adversarial and reconstruction loss.
6. Save a checkpoint for each run in `outputs/checkpoints`, including trained parameters, training losses, RMSE histories, validation split information, and the resolved configuration.
7. Evaluate each checkpoint on the holdout subset by computing generator RMSE in the wavelet domain and comparing it against the identity-style baseline that reuses the input tensor as a proxy prediction.
8. Write per-run summary rows to `outputs/experiment_summary.csv` so the three horizons can be compared side by side.
9. Generate report-oriented PNG figures for each run, including tensor visualizations, qualitative comparison panels, and training-diagnostics plots.
10. Copy those figures into `reports/figures` so the LaTeX report can reference stable, local assets.

Taken together, these steps provide a complete workflow for data preparation, model training, checkpointing, evaluation, and figure generation for the three WaveGAN forecasting horizons studied here.

References

[1] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017.

[2] Yui Chun Leung. Matlab-gan. `https://github.com/zcemycl/Matlab-GAN.git`, 2020.

[3] Sam McHardy. python-binance. `https://github.com/sammchardy/python-binance.git`, 2021.

[4] Ivan W. Selesnick, Richard G. Baraniuk, and Nick C. Kingsbury. The dual-tree complex wavelet transform. *IEEE Signal Processing Magazine*, 2005.