AI Edge

Harness AI for Smarter, Data-Driven Decisions

AI Edge Indicators Documentation

Build: 5b20977 / 2025-09-12T19:03:36.117Z / production

Lorentzian Classification Indicator

What you'll learn

This comprehensive guide will walk you through the intricacies of the Lorentzian Classification Indicator, a powerful tool for technical analysis in TradingView.

Introduction

Whether it's the frozen milliseconds of panic after a black swan event, when time seems to grind to a halt, or the face-melting euphoria of a bull market's final surge when time accelerates into a blur -- it is in these moments of extremes that we may catch a glimpse of a profound truth: markets don't exist in flat, Euclidean space.

Why This Matters

Just as massive objects bend the fabric of spacetime, significant economic events warp the geometry of markets. These distortions ripple through what we might call the fabric of "price-time," transforming the very space in which markets exist. Yet traditional technical analysis, built for a flat and uniform world, continues to assume that past and present can be compared directly. In doing so, it makes a devastating mistake: treating curved space as if it were flat—leaving traders vulnerable during those turbulent moments when market geometry warps most violently, when accurate analysis is crucial to navigating market extremes.

How Lorentzian Classification Works

Drawing from the mathematics of curved space in theoretical physics, Lorentzian Classification charts a new course through warped market geometry. Where traditional indicators assume flat space, we embrace the curve. Where they measure linear distance, we measure along geodesics. Where they break down, we adapt. At the heart of this approach lies the Lorentzian distance metric—a sophisticated mathematical tool born from relativity theory. Paired with a novel nearest neighbors algorithm that ensures chronologically diverse reference points, this framework adeptly navigates the complex curves and distortions of market space, surpassing the limitations of traditional analysis.

What This Means for Traders

From this curved space emerges a revolutionary approach to understanding markets—one that not only survives moments of extreme volatility but also harnesses it into a source of valuable analytical insights. Through classification scores, we quantify the unquantifiable. Through kernel regression, we visualize the invisible. Through adaptive filtering, we navigate the unnavigable. Yet this framework isn't meant to be a crystal ball gazing into future price action. Rather, it serves as a sophisticated lens through which traders can finally see markets as they truly are: curved spaces where fear and greed bend the very fabric of price and time.

What This Guide Covers

The following guide is a high-level overview of the major sections you'll find, starting with the Premium Version enhancements. Each section will build on the concepts of curvature-aware market analysis, providing both conceptual depth and practical techniques.

ACHIEVEMENT

Now ranking among the Top 30 indicators of all time with 20k boosts, Lorentzian Classification has evolved into one of TradingView's most influential indicators.

Understanding Core Concepts

Imagine entering a market suddenly shaken by unexpected news—where time seems distorted, fundamentals are sidelined, and every price movement feels exaggerated. Traditional technical tools, which assume that each moment is comparable in a straightforward, linear way, often struggle in these extreme conditions. Lorentzian Classification offers a fresh perspective, rooted in the concept that markets, much like spacetime, can bend under pressure. By acknowledging this curvature, the indicator reveals patterns that conventional, flat, Euclidean methods might overlook.

Lorentzian Space & Distance

Why "Warping" Matters

Traditional indicators often rely on Euclidean distance—a straight-line measure of how close two price points are. But real markets aren't "flat." Major events like FOMC announcements or Black Swan shocks can drastically "bend" market behavior, creating pockets of extreme volatility or momentum.

> **KEY DIFFERENCE**
>
> Unlike traditional indicators using Euclidean distance (straight-line), Lorentzian distance accounts for market "warping" effects near significant events, treating distance less like a ruler and more like a stretchy piece of fabric.

Adaptive Analysis

This resilience to sudden market shifts makes Lorentzian Classification especially robust around chaotic events where normal analysis might break down. The distance metric adapts to:

- Sudden volatility spikes
- Momentum surges
- Regime changes
- Market structure shifts

Neighborhoods & Classification

Finding Historical Analogs

The indicator works by identifying "neighborhoods" of similar historical price action. Think of it as:

- Finding similar market situations in history
- Analyzing what typically happened next
- Preventing temporal clustering bias through diverse neighbor selection

Advanced Pattern Recognition

Instead of clustering around just the most recent data, it casts a wide historical net. By maintaining chronological diversity (rather than just "recentness"), Lorentzian Classification mitigates the bias of short-term memory, delivering context from a variety of past market regimes.

Another example of advanced pattern recognition used by the Lorentzian Classification Indicator is its ability to identify "first pullbacks" within a trend. Recognizing these initial retracements can be crucial not only for finding favorable entry points but also for risk mitigation. By entering during a pullback, traders reduce the risk of buying at the peak of a trend, which could lead to unfavorable positions if the trend then reverses. These pullbacks are akin to natural phenomena where a brief counter-movement is followed by a continued advance in the initial trend direction. By analyzing price action, the indicator captures instances where a trend resumes after a temporary setback, offering actionable insights for traders aiming to capitalize on these occurrences while managing risk effectively.

Voting System Mechanics

Each neighbor "votes" on what happened next in its historical situation. These votes then aggregate into a single directional bias—bullish, bearish, or somewhere in-between. This approach:

- Channels wisdom from multiple time periods
- Reduces overfitting risk
- Provides more robust signals
- Adapts to different market conditions

Bar Scores & Signals

Intuitive Numeric Scale

With 8 neighbors, each bar receives a numerical score:

Score	Meaning	Interpretation
+8	Strongly bullish	All neighbors agree bullish

Score	Meaning	Interpretation
+4 to +7	Moderately bullish	Majority bullish consensus
-3 to +3	Neutral/Mixed	Market in transition or conflict
-4 to -7	Moderately bearish	Majority bearish consensus
-8	Strongly bearish	All neighbors agree bearish

VISUAL FEEDBACK

Scores are visualized through color gradients on the bars, providing intuitive market sentiment readings. Strong reds, greens, or neutrals give you a quick read on sentiment strength.

Actionable Insights

- **Extreme Scores**: Suggest clear directional trades
- **Middling Values**: Hint at uncertainty or consolidation
- **Score Transitions**: Can signal momentum shifts
- **Pattern Confirmation**: Verify market remains in harmony with historical analogs

Kernel Regression

Beyond Traditional Moving Averages

Traditional moving averages apply a fixed formula, whether exponential or simple, often overlooking sudden regime changes. Lorentzian Classification's kernel regression deploys the Rational Quadratic Kernel, a more agile method that adapts its weighting based on real-time market conditions.

Advanced Capabilities

The kernel regression functions as a "moving average on steroids" by:

- Smoothing out noise while maintaining sensitivity
- Adapting to volatility changes
- Tracking momentum shifts
- Providing dynamic support/resistance levels
- Generating early warning signals through slope changes

ⓘ **ADVANCED MOVING AVERAGE**

Think of it as a "moving average on steroids" using the Rational Quadratic Kernel for more nuanced trend analysis. It smooths out noise yet stays sensitive to shifts in volatility, momentum, and direction, capturing subtle changes faster than standard MAs.

Practical Applications

Market Analysis

- Quick visual assessment through color gradients
- Early detection of trend changes
- Identification of key support/resistance levels
- Recognition of market regime shifts

Trading Strategy Integration

- Entry signal confirmation
- Exit timing optimization
- Risk management framework
- Multi-timeframe analysis support

Summary

From warped distances that stay robust during volatility spikes to bar scores that reveal crowd consensus and a kernel regression that "breathes" with the market, Lorentzian Classification weaves a richer, more adaptable tapestry of insights. By viewing market data through this curved-space perspective, traders can better anticipate the twists and turns that define today's rapidly shifting financial landscape.

Premium Features

Custom Sources Architecture

Lorentzian Classification Premium v3.0 introduces a powerful custom sources architecture that revolutionizes how you can analyze market data:

- Process up to 5 external data sources simultaneously
- Leverage up to 10 dimensional features for analysis
- Perform multi-timeframe data processing

Feature Engineering

The enhanced feature engineering system allows you to:

- Integrate custom indicators from TradingView
- Apply advanced normalization algorithms
- Cross-reference multiple data sources for improved accuracy

Dynamic Kernel Regression Ribbon

The kernel regression ribbon provides real-time market analysis through:

Visual Components

- Real-time kernel estimate visualization
- Color-coded prediction strength indicators
- Automatic support/resistance identification

Analysis Capabilities

- Instant trend strength feedback
- Early reversal warning system
- Market regime change detection

Mean Reversion Detection System

The mean reversion system represents a major advancement in identifying market reversals. It combines sophisticated mathematical models with multi-factor confirmation to generate reliable signals.

Implementation Features

Designed for both scalpers and swing traders, the system offers:

- Precise scalping optimization
- Comprehensive swing trading capabilities
- Customizable alert conditions
- Dynamic exit point synchronization

Signal Backtesting Framework

Signal Range (-5 to +5)

Signal	Meaning	Application
±5	Strong reversals	Highest probability mean reversion opportunities
±4	Regular reversals	Standard mean reversion entries
±3	First pullbacks	Optimal trend-following entries
±1,±2	Trade entries/exits	Position management signals

Implementation Features

- **Comprehensive Historical Analysis**: Use PineScript's Strategy Tester or an external Backtest Adapter to validate signals over large data samples.
- **Custom Alerts & Automation**: Tie specific numeric ranges to TradingView alerts, and even integrate with auto-trading bots.
- **Performance Tracking**: The built-in stats panel can guide real-time calibration. Though not a replacement for thorough backtesting, it offers immediate feedback on each signal's effectiveness.

> ⓘ **PRO TIP**
>
> Combine multiple signal types to create sophisticated trading strategies. For example, use ±5 signals for mean reversion trades while using ±3 signals to enter trending markets.

Symbols and Alerts Legend

Introduction — What this table is for

Use the table below as a quick-reference legend to every visual cue produced by the Lorentzian Classification Premium indicator. Each row groups a symbol by its trading Category (Entry/Exit, Continuation, Mean-Reversion, Deviation, or Context) and shows:

- an inline SVG icon that mirrors exactly what you see on the chart,
- a plain-language description of how the symbol is drawn and what it means in live trading,
- the Alert Identifier—the precise condition name you'll select in TradingView's Create Alert dialog to be notified when that event occurs.

If you ever wonder "What does this arrow/triangle/dot mean?" or "Which alert should I pick for it?", this table is the authoritative answer.

Category	Symbol Name	Symbol	Additional Description	Meaning	Alert Identifier
Entry / Exit	**Buy**	⬆	Teal arrow plotted **below** the candle	Opens a long position when a bullish move is expected	**Open Long ▲**
	Sell	⬇	Red arrow plotted **above** the candle	Opens a short position when a bearish move is expected	**Open Short ▼**
	Stop Buy Cross	✕	Small green "X" at the exit price	Closes a long after its holding period or on a dynamic exit	**Close Long ▲**

Category	Symbol Name	Symbol	Additional Description	Meaning	Alert Identifier
	Stop Sell Cross	✕	Small red “X” at the exit price	Closes a short after its holding period or on a dynamic exit	**Close Short ▼**
Continuation	Pullback Up	▲	Green triangle **below** the bar	First-pullback continuation entry in an up-trend	**Upward First Pullback Detected**
	Pullback Down	▼	Red triangle **above** the bar	First-pullback continuation entry in a down-trend	**Downward First Pullback Detected**
Mean-Reversion	Standard Up	🔄	Circular arrow printed **below** the bar	Moderate oversold deviation likely to bounce	**Upward Mean Reversion Detected**
	Standard Down	🔄	Circular arrow printed **above** the bar	Moderate overbought deviation likely to pull back	**Downward Mean Reversion Detected**
	Strong Up	⬆️	Bold up-arrow below the bar	Extreme downside deviation—high-conviction bounce signal	**Strong Upward Mean Reversion Detected**
	Strong Down	⬇️	Bold down-arrow above the bar	Extreme upside deviation—	**Strong Downward Mean**

Category	Symbol Name	Symbol	Additional Description	Meaning	Alert Identifier
				high-conviction pullback signal	**Reversion Detected**
Deviation	Deviation Dot Below	●	Green dot when price is well **below** the kernel (optional)	Custom alert: price far below mean, possible snap-back	**Price Far Below Kernel**
	Deviation Dot Above	●	Red dot when price is well **above** the kernel (optional)	Custom alert: price far above mean, possible snap-back	**Price Far Above Kernel**
Context	Vote-Score Label	–8 … +8	Number printed above/below candle, tinted green/red/gray	Shows the model's raw confidence for that bar	N/A
	Confidence Gradient Bar	*(bar)*	Candle body tinted green/red/gray with strength gradient	Quick visual of bullish, bearish, or neutral sentiment	N/A

Notes on using the Alert Identifier column

To create an alert corresponding to any of these symbols:

1. Open TradingView's Alerts → Create Alert while the indicator is on your chart.
2. In the Condition dropdown select Lorentzian Classification Premium, then choose the precise text shown in the Alert Identifier column (e.g., "Open Long ▲" or "Price Far Above Kernel").
3. Set Trigger to "Once per bar close" so the alert fires just after the candle that created the symbol has completed.

4. Save the alert—TradingView will now notify you whenever that symbol appears, subject only to the platform’s normal near-real-time delivery.

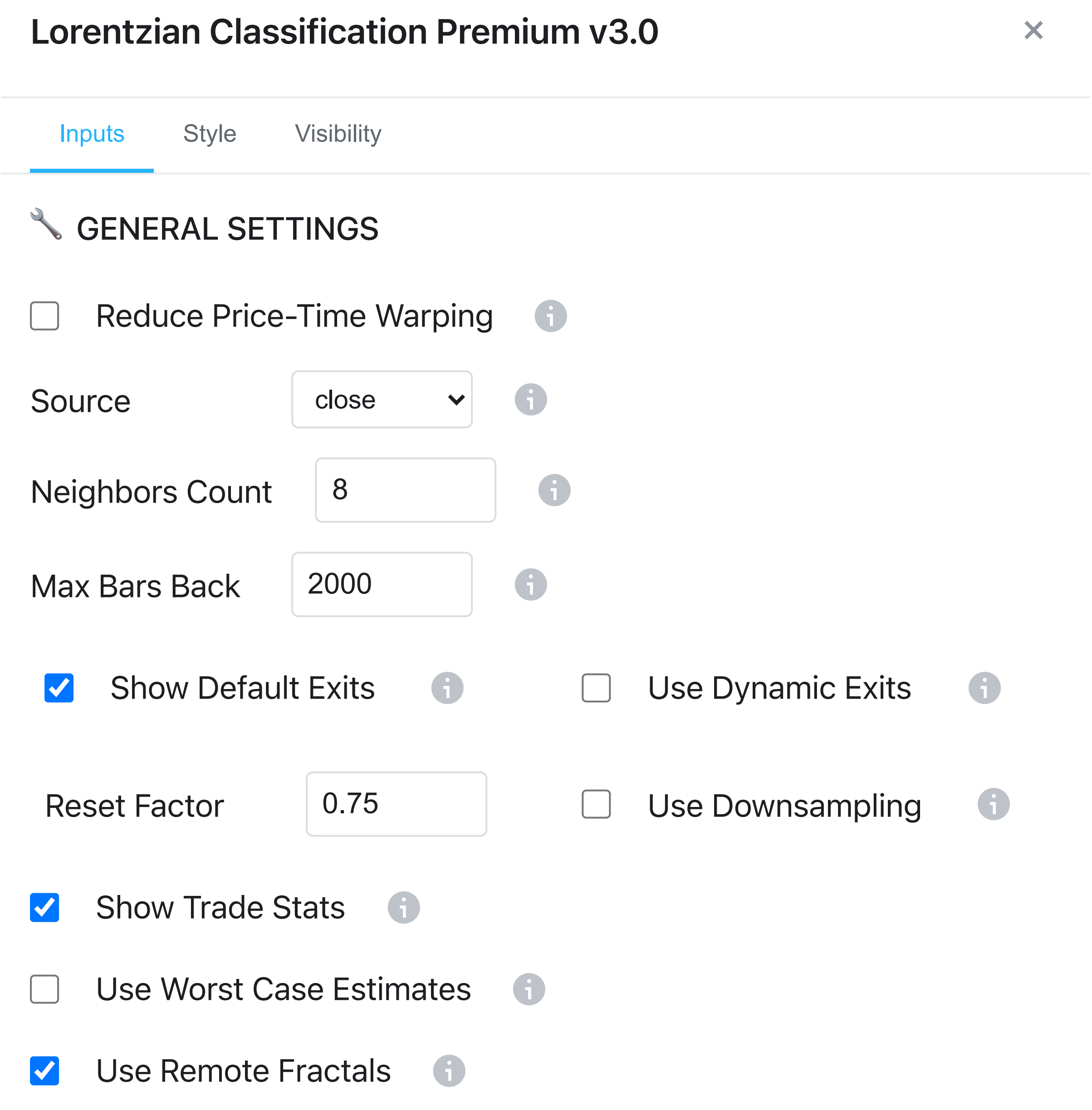
Rows marked N/A have no dedicated alert; they are visual aids only.

General Settings

These settings form the baseline configuration for the Lorentzian Classification model. They control aspects like data sources, how far into the past the model looks, how it calculates distances, how trades are displayed and timed, and the complexity of the training logic.

Interactive Settings Explorer

Below is an interactive exploration of the General Settings interface. Click any setting to instantly view its detailed documentation, default value, and purpose - a quick alternative to scrolling through the full reference below:



Lorentzian Classification Premium v3.0

🔧 GENERAL SETTINGS

□ Reduce Price-Time Warping

Source: close

Neighbors Count: 8

Max Bars Back: 2000

☑ Show Default Exits

□ Use Dynamic Exits

Reset Factor: 0.75

□ Use Downsampling

☑ Show Trade Stats

□ Use Worst Case Estimates

☑ Use Remote Fractals

Detailed Settings Reference

Core Configuration Settings

Reduce Price-Time Warping

Default: `false`

PURPOSE

Minimizes distortions in price-time relationships during pattern matching.

IMPACT

Affects how the model identifies and weighs historical patterns.

RECOMMENDATIONS

- Enable for markets with high volatility
- Disable for range-bound markets
- Test both settings in your specific market context

Source

Default: `close`

PURPOSE

Determines which price point of each bar is used for calculations (close, open, high, or low).

IMPACT

The choice of price source can significantly affect how the indicator interprets market movements and generates signals.

RECOMMENDATIONS

- Use 'close' for most reliable signals
- Consider 'high/low' for range-based strategies
- Match source to your trading timeframe

BEST PRACTICE

Stick with the default `close` source unless you have a specific reason to change it. This ensures optimal compatibility with the model's core algorithms.

Neighbor Configuration

Neighbors Count

Default: `8`

PURPOSE

Sets the number of nearest neighbors used in the classification algorithm. More neighbors can provide smoother signals but may reduce sensitivity.

IMPACT

Higher values create more stable but slower signals, while lower values increase sensitivity but may generate more noise.

RECOMMENDATIONS

- Start with the default of 8 neighbors
- Increase for less volatile markets
- Decrease for more responsive signals

Max Bars Back

Default: `2000`

PURPOSE

Defines the maximum number of historical bars used for calculations. Affects both accuracy and performance.

IMPACT

Larger values provide more historical context but require more computational resources.

RECOMMENDATIONS

- Use at least 1000 bars for reliable signals

- Adjust based on your trading timeframe
- Consider your platform's performance capabilities

Exit Settings

Show Default Exits

Default: `true`

PURPOSE

Displays the indicator's built-in exit signals based on kernel regression analysis.

IMPACT

Helps identify potential exit points for trades, improving risk management.

RECOMMENDATIONS

- Keep enabled for complete trading signals
- Use in conjunction with your own exit rules
- Compare with dynamic exits for confirmation

Use Dynamic Exits

Default: `false`

PURPOSE

Calculates exit points dynamically based on market volatility and trend strength.

IMPACT

Provides more adaptive exit signals that adjust to changing market conditions.

RECOMMENDATIONS

- Enable for volatile markets
- Combine with fixed exits for confirmation
- Monitor effectiveness in your timeframe

Advanced Configuration

Reset Factor

Default: `0.75`

PURPOSE

Controls how aggressively the algorithm resets its internal state when market conditions change significantly.

IMPACT

Higher values make the indicator more responsive to changes but may increase false signals.

RECOMMENDATIONS

- Lower values for trending markets
- Higher values for ranging markets
- Adjust based on false signal frequency

TECHNICAL DETAIL

Lower reset factors create more selective neighbor choices, while higher values are more lenient in maintaining historical relationships.

Use Downsampling

Default: `false`

PURPOSE

Reduces data resolution to improve performance while maintaining signal quality.

IMPACT

Can significantly improve performance on larger datasets with minimal impact on accuracy.

RECOMMENDATIONS

- Enable for very large datasets
- Use when experiencing performance issues

- Consider disabling for precise entry/exit timing

Performance Monitoring

Show Trade Stats

Default: `true`

PURPOSE

Displays real-time statistics about the indicator's performance and trade signals.

IMPACT

Provides immediate feedback on the indicator's effectiveness and helps in strategy refinement.

RECOMMENDATIONS

- Keep enabled during strategy development
- Use to validate signal quality
- Monitor for strategy optimization

Use Worst Case Estimates

Default: `false`

PURPOSE

Calculates signals using conservative estimates to reduce false positives.

IMPACT

Generates fewer but potentially more reliable signals.

RECOMMENDATIONS

- Enable for risk-averse strategies
- Use in volatile market conditions
- Combine with other confirmation tools

Pattern Recognition

Use Remote Fractals

Default: `true`

PURPOSE

Incorporates fractal pattern analysis from higher timeframes for signal generation.

IMPACT

Improves signal quality by considering larger market structures.

RECOMMENDATIONS

- Keep enabled for trend following
- Especially useful in higher timeframes
- Monitor impact on signal timing

Default Configuration Rationale

The default settings are carefully chosen to provide:

- Simplified initial setup
- Clear base exit logic
- Rich historical context
- Balanced performance metrics
- Optimistic preview mode

> ⓘ **WHY THESE DEFAULTS?**
>
> Default values prioritize ease of understanding while maintaining the indicator's core strengths. They provide a solid foundation for both learning and customization.

Summary

These general settings provide the foundation for Lorentzian Classification's behavior. While the defaults offer a balanced starting point, understanding each setting's purpose allows for precise tuning to match your trading style and market conditions.

Try experimenting with the interactive demo above to see how different settings affect the configuration. Each setting has detailed documentation that appears when you hover over it in the interface.

Example Configurations

Quick & Simple Setup:

- **Reduce Price-Time Warping:** False - *Reasoning: Maintains the indicator's core sensitivity to market volatility, allowing it to react quickly to sudden price changes.*
- **Source:** close - *Reasoning: Uses the standard closing price for broad compatibility, ensuring the indicator works consistently across different assets and exchanges.*
- **Neighbors Count:** 8 - *Reasoning: Provides a balanced number of neighbors for classification, offering a good trade-off between responsiveness and noise reduction.*
- **Max Bars Back:** 2000 - *Reasoning: Offers a reasonable historical context without excessive computation, allowing the indicator to analyze a sufficient amount of past data without slowing down performance.*
- **Show Exits:** False - *Reasoning: Simplifies the chart by hiding exit signals, reducing visual clutter and focusing on entry points.*
- **Use Dynamic Exits:** False - *Reasoning: Uses a fixed exit strategy for simplicity, making it easier to understand and manage trades.*
- **Use Remote Fractals:** True - *Reasoning: Leverages a broader historical context for pattern recognition, potentially identifying more reliable signals.*
- **Show Trade Stats:** True - *Reasoning: Provides basic performance feedback, allowing users to quickly assess the effectiveness of the configuration.*
- **Use Worst Case Estimates:** False - *Reasoning: Shows a more optimistic view of potential performance, encouraging experimentation and exploration. This configuration is designed for New to ML/Trading Beginners seeking an easy starting point. It uses default settings for most parameters, providing a balanced configuration for general use and easy exploration of the indicator's basic functionality. Remote Fractals are enabled to provide a broader historical context, while keeping other settings at their defaults allows for a less overwhelming initial experience.*

Robust Trend Confirmation:

- **Reduce Price-Time Warping:** True - *Reasoning: Reduces sensitivity to short-term volatility, emphasizing longer trends and providing more stable signals.*
- **Source:** close - *Reasoning: Uses the standard closing price for broad compatibility, ensuring consistent results across different assets and exchanges.*

- **Neighbors Count:** 12 - *Reasoning: Increases the number of neighbors for more robust trend confirmation, reducing the impact of outliers and improving signal reliability.*
- **Max Bars Back:** 3000 - *Reasoning: Provides a richer historical context for trend analysis, allowing the model to learn from a wider range of market conditions and identify more reliable trends.*
- **Show Exits:** True - *Reasoning: Displays exit signals for clear trade management, enabling users to visualize potential exit points and manage risk effectively.*
- **Use Dynamic Exits:** True - *Reasoning: Adapts exit points to changing market conditions, potentially improving profitability and reducing losses.*
- **Use Remote Fractals:** False - *Reasoning: Focuses on recent price action for trend detection, reducing the influence of distant historical patterns that may no longer be relevant.*
- **Show Trade Stats:** True - *Reasoning: Provides performance feedback to monitor the effectiveness of the configuration and track key metrics like win rate and drawdown.*
- **Use Worst Case Estimates:** True - *Reasoning: Provides a conservative view of potential performance, aligning with risk-averse strategies and ensuring a realistic assessment of potential losses. This configuration is designed for Fixed Income/Conservative Traders and Backtest Optimizers prioritizing robust trend confirmation. It reduces price-time warping for more Euclidean character, increases Neighbors Count and Max Bars Back for robust trend confirmation, and enables Dynamic Exits and Worst Case Estimates for conservative backtesting. The focus is on minimizing risk and maximizing the reliability of trend-following signals, making it suitable for long-term investors and those with a low-risk tolerance.*

Optimized for Scalping:

- **Reduce Price-Time Warping:** False - *Reasoning: Maintains sensitivity to short-term price fluctuations, crucial for identifying quick scalping opportunities.*
- **Source:** close - *Reasoning: Uses the standard closing price for broad compatibility, ensuring consistent results across different assets and exchanges.*
- **Neighbors Count:** 5 - *Reasoning: Reduces the number of neighbors for faster signal generation, essential for quick entries and exits in fast-moving markets.*
- **Max Bars Back:** 1000 - *Reasoning: Minimizes computational load and emphasizes recent price action, improving responsiveness and reducing lag.*
- **Show Exits:** False - *Reasoning: Simplifies the chart for quick decision-making, reducing visual clutter and allowing for faster reactions to market changes.*
- **Use Dynamic Exits:** True - *Reasoning: Adapts exit points to fast-moving market conditions, allowing for quicker profit-taking or loss-cutting and maximizing efficiency.*
- **Use Remote Fractals:** True - *Reasoning: Leverages a broader historical context for pattern recognition, potentially identifying subtle scalping opportunities that might be missed with a*

shorter lookback period.

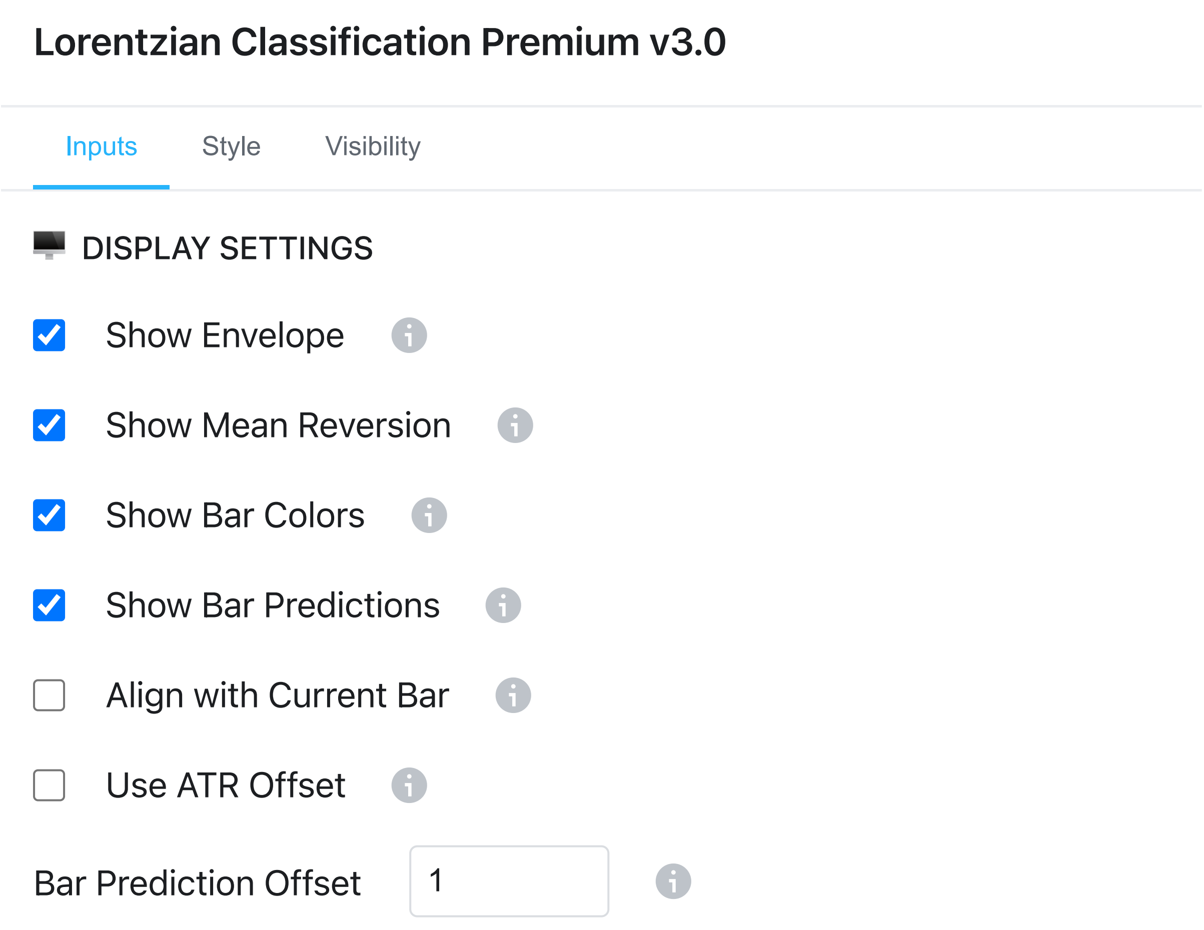
- **Show Trade Stats:** True - *Reasoning: Provides performance feedback to monitor the effectiveness of the configuration and track key metrics like win rate and profit factor.*
- **Use Worst Case Estimates:** False - *Reasoning: Prioritizes speed over conservative backtesting, reflecting the aggressive nature of scalping and focusing on potential gains rather than worst-case scenarios. This configuration is tailored for Performance Tweakers and Reactive Intraday Traders focused on low-latency scalping. It reduces Neighbors Count and Max Bars Back to minimize computational load and increase responsiveness for scalping. Dynamic Exits are enabled for quicker exits, while Worst Case Estimates are disabled to prioritize speed over conservative backtesting. The goal is to generate frequent, timely signals for quick profit opportunities, even if it means accepting a higher risk of false signals.*

Display Settings

These settings control the visual representation and interface elements of the Lorentzian Classification model. They determine how predictions are displayed, how signals are positioned, and how various visual elements interact to create a cohesive trading interface.

Interactive Settings Explorer

Below is an interactive exploration of the Display Settings interface. Click any setting to instantly view its detailed documentation, default value, and purpose - a quick alternative to scrolling through the full reference below:



Lorentzian Classification Premium v3.0

DISPLAY SETTINGS

☑ Show Envelope

☑ Show Mean Reversion

☑ Show Bar Colors

☑ Show Bar Predictions

□ Align with Current Bar

□ Use ATR Offset

Bar Prediction Offset: 1

Detailed Settings Reference

Core Visual Elements

Show Envelope

Default: `true`

PURPOSE

Visualizes envelope bands on the chart that interact with the Lorentzian Classification indicator's kernel regression, serving as a confluence tool for understanding price action and potential mean reversion points.

IMPACT

The envelope bands provide visual context for market volatility and potential support/resistance levels, helping identify overbought and oversold conditions.

RECOMMENDATIONS

- Keep enabled for comprehensive market analysis
- Use band interactions to confirm reversals
- Monitor color changes for trend strength

Show Mean Reversion

Default: `true`

PURPOSE

Displays advanced mean reversion detection signals that adapt to market conditions and evolve with accumulated data, particularly useful for identifying potential market reversals.

IMPACT

Provides actionable insights during intra-bar analysis and helps catch quick scalping opportunities through adaptive classification signals.

RECOMMENDATIONS

- Use for contrarian trading strategies
- Monitor signal evolution over time
- Combine with envelope bands for confirmation

BEST PRACTICE

> Keep both envelope and mean reversion displays enabled for maximum insight. The envelope provides context while mean reversion signals highlight potential reversals.

Signal Representation

Show Bar Colors

Default: `true`

PURPOSE

Colors each chart bar according to the model's prediction (green for bullish, red for bearish, gray for neutral), providing immediate visual feedback on market sentiment.

IMPACT

Enhances trend recognition and helps maintain focus on larger market dynamics rather than short-term price fluctuations.

RECOMMENDATIONS

- Use in conjunction with bar predictions
- Monitor color transitions for trend changes
- Consider disabling if chart becomes too busy

Show Bar Predictions

Default: `true`

PURPOSE

Displays the model's raw prediction score as a numeric label on each bar, providing detailed insight into the strength and direction of predictions.

IMPACT

Enables precise assessment of model confidence and helps in timing trading decisions based on prediction strength.

RECOMMENDATIONS

- Use alongside bar colors for complete analysis
- Monitor prediction trends over time

- Consider your chart's visual clarity

TECHNICAL DETAIL

Bar colors and predictions work together to provide both immediate visual feedback and detailed numerical insights into the model's confidence levels.

Signal Positioning

Align Signal with Current Bar

Default: `false`

PURPOSE

Controls the timing offset for plotted signals, determining whether they appear on the last closed bar or the current forming bar.

IMPACT

Affects the historical accuracy and real-time responsiveness of signal placement, impacting backtesting results and live trading execution.

RECOMMENDATIONS

- Keep disabled for accurate historical analysis
- Enable only for real-time strategy testing
- Consider backtesting implications

Use ATR Offset

Default: `true`

PURPOSE

Dynamically adjusts the position of prediction values and labels based on the Average True Range (ATR), ensuring clear visibility regardless of market volatility.

IMPACT

Maintains chart readability by preventing label overlap during high volatility and keeping labels appropriately positioned in calmer markets.

RECOMMENDATIONS

- Keep enabled for automatic label positioning
- Use with bar prediction offset for fine-tuning
- Monitor label placement in different conditions

Default Configuration Rationale

The default display settings are optimized to provide:

- Maximum signal clarity
- Intuitive visual hierarchy
- Reduced chart clutter
- Accessible color schemes
- Balanced alert feedback

> ⓘ **WHY THESE DEFAULTS?**
>
> Default values prioritize ease of understanding while maintaining the flexibility needed for custom trading setups. They provide a clean, professional look that works well across different chart types and timeframes.

Visual Element Relationships

Understanding how different visual elements interact is key to optimizing your display configuration:

- **Envelope + Mean Reversion**: Provides context for potential reversals
- **Bar Colors + Predictions**: Offers both quick-glance and detailed insights
- **ATR Offset + Signal Placement**: Ensures clear signal visibility in varying market conditions
- **Label Size + Line Thickness**: Creates visual hierarchy for different information types

Summary

These display settings form the interface between the model's analysis and your trading decisions. While the defaults provide a professional and clear starting point, understanding each setting's purpose allows you to create a personalized visual environment that enhances your trading workflow.

Try experimenting with the interactive demo above to see how different settings affect the visualization. Each setting has detailed documentation that appears when you click on it in the interface.

Example Configurations

Clean & Minimalist Display:

- **Show Envelope:** False - *Reasoning: Reduces visual clutter, focusing on essential signals.*
- **Show Mean Reversion:** False - *Reasoning: Further declutters chart, removing less critical reversal signals.*
- **Show Bar Colors:** False - *Reasoning: Removes bar color gradients for a cleaner price action view.*
- **Show Bar Predictions:** False - *Reasoning: Hides numerical prediction values, minimizing on-chart text.*
- **Use ATR Offset:** False - *Reasoning: Positions signals directly on bars for minimal visual distraction.*
- **Align Signal with Current Bar:** False - *Reasoning: Plots signals on bar close for historical signal stability. This configuration is designed for Visual Confluence Traders and Performance Tweakers who prioritize a clean, minimalist chart and maximum performance. By disabling most visual elements, it reduces chart clutter and minimizes indicator load, focusing solely on the essential buy and sell signals. This setup is ideal for traders who rely on other indicators for confluence and want to keep the Lorentzian Classification indicator as a secondary confirmation tool without overwhelming the chart.*

Enhanced Signal Visibility Display:

- **Show Envelope:** True - *Reasoning: Provides visual context for trend and potential reversals.*
- **Show Mean Reversion:** True - *Reasoning: Highlights potential mean reversion opportunities.*
- **Show Bar Colors:** True - *Reasoning: Offers quick visual assessment of trend strength.*
- **Show Bar Predictions:** True - *Reasoning: Provides detailed numerical insights into model confidence.*
- **Use ATR Offset:** True - *Reasoning: Ensures signals are clearly visible above/below bars, even in volatile conditions.*
- **Align Signal with Current Bar:** True - *Reasoning: Aligns signals to the current bar for immediate visual feedback (note: signals are still based on closed bar data). This configuration*

is designed for New to ML/Trading Beginners and Visual Confluence Traders who benefit from maximum visual feedback and clear signal representation. By enabling all visual elements, it provides a comprehensive view of the model's analysis, with clear color gradients, numerical predictions, and contextual elements like the envelope and mean reversion signals. This setup is ideal for users who are still learning the indicator and want to maximize visual information for better understanding and decision-making.

Trend Focus Display:

- **Show Envelope:** True - *Reasoning: Provides visual context for the prevailing trend.*
- **Show Mean Reversion:** False - *Reasoning: Removes mean reversion signals to focus exclusively on trend-following signals.*
- **Show Bar Colors:** True - *Reasoning: Color-codes bars to quickly visualize trend direction and strength.*
- **Show Bar Predictions:** False - *Reasoning: Hides numerical predictions to declutter the chart and emphasize visual trend signals.*
- **Use ATR Offset:** True - *Reasoning: Ensures trend signals are clearly visible.*
- **Align Signal with Current Bar:** False - *Reasoning: Plots signals on bar close for historical signal stability and trend confirmation. This configuration is tailored for Visual Confluence Traders who primarily focus on trend-following strategies. By enabling the envelope and bar colors while disabling mean reversion signals and bar predictions, it emphasizes the visual representation of the trend direction and strength. The kernel estimate and bar colors provide a clear and uncluttered view of the prevailing trend, while the signals are positioned on bar close for historical signal stability and trend confirmation. This setup is ideal for traders who use Lorentzian Classification as a confluence tool to confirm trend-based trading decisions.*

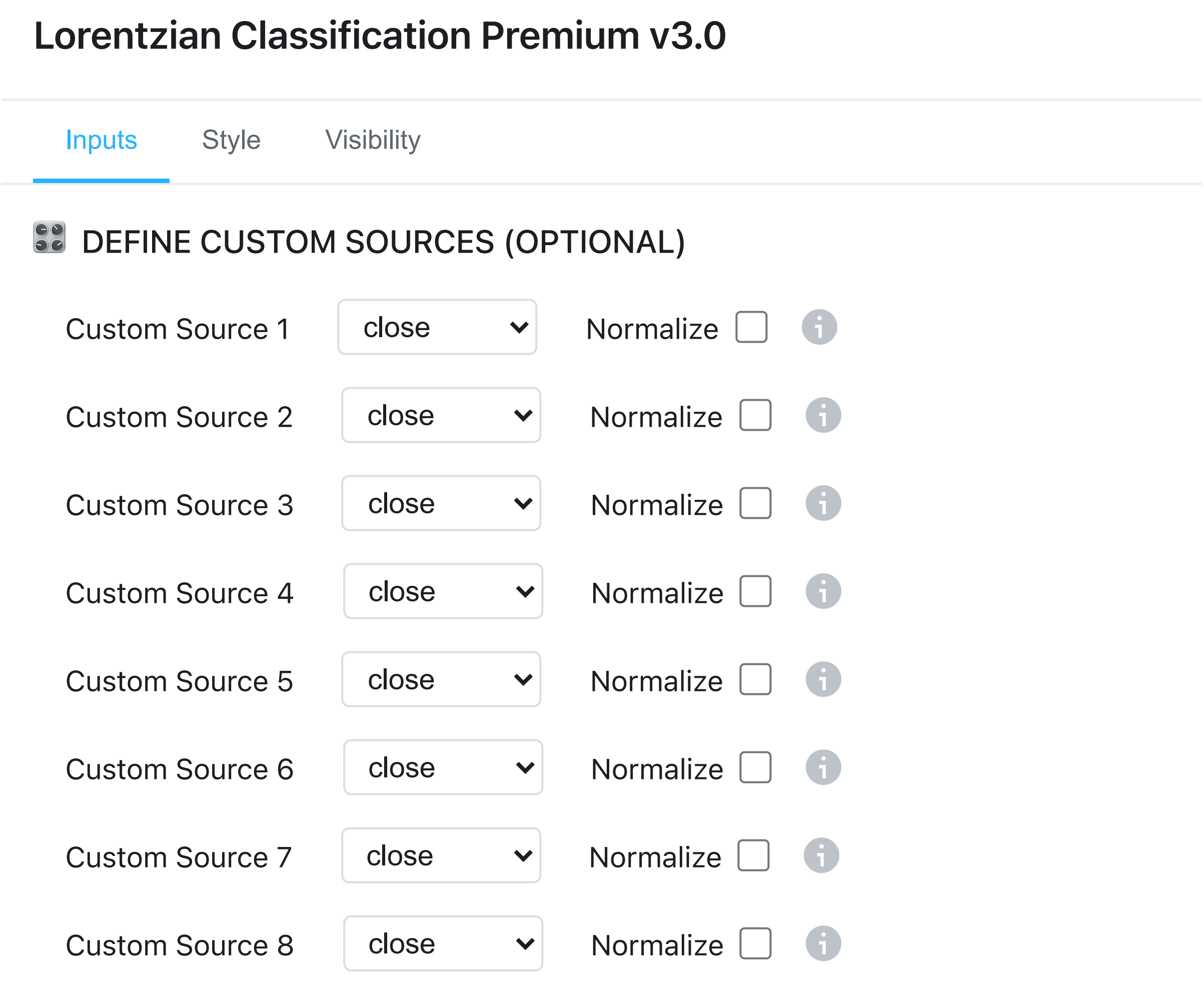
Custom Sources

Custom Sources in the Lorentzian Classification indicator enable you to integrate external or specialized data streams—beyond standard price, volume, and built-in technical indicators—directly into the machine learning model. By doing so, you expand the "feature space" available to the Lorentzian Classification algorithm, potentially increasing its ability to capture complex market behavior across different market regimes (ranging, trending, volatile, or quiet).

In essence, a "Custom Source" is any numeric series available in your TradingView chart environment. You might choose a data series from a proprietary script, a specialized volume indicator, a fundamental metric, or anything else that outputs a numerical value each bar. These optional inputs can then be combined and normalized to act as additional features for the Lorentzian Classification model.

Interactive Settings Explorer

Below is an interactive exploration of the Custom Sources interface. Click any setting to instantly view its detailed documentation, default value, and purpose:



Lorentzian Classification Premium v3.0

DEFINE CUSTOM SOURCES (OPTIONAL)

Custom Source 1	close	Normalize	□
Custom Source 2	close	Normalize	□
Custom Source 3	close	Normalize	□
Custom Source 4	close	Normalize	□
Custom Source 5	close	Normalize	□
Custom Source 6	close	Normalize	□
Custom Source 7	close	Normalize	□
Custom Source 8	close	Normalize	□

Custom Source 9 close Normalize □

Custom Source 10 close Normalize □

Why Custom Sources Matter

1. **Expanded Dimensionality**
   Originally, Lorentzian Classification was constrained by a limited set of built-in indicators (e.g., RSI, ADX). The ability to add custom data streams broadens the number of features from a handful to as many as ten. This additional dimensionality enhances the model's capacity to capture nuanced patterns that might otherwise go unnoticed.

2. **Tailored Analysis**
   Traders often have personal theories or preferred indicators for specific market conditions, such as using WaveTrend 3D for multi-timeframe oscillations or a specialized volume metric for capitulation events. Custom Sources allow you to weave these exact data streams into the classification process, creating a model better aligned with your trading style or hypotheses about market behavior.

3. **Adaptability**
   Markets evolve; what works in one environment can become less effective in another. By letting you insert and remove data series at will, Custom Sources facilitate rapid experimentation. If you suspect a new indicator might capture emerging market conditions, you can test it quickly without rewriting the entire code base. This flexible design encourages continuous iteration and refinement of your trading model.

4. **Diverse Thematic Focus**
   One user might want to concentrate on momentum (via multiple oscillators), another might want to incorporate macro-level data or order-flow metrics. The "theme" of your Lorentzian Classification system can shift in response to your research or changing market dynamics. With Custom Sources, you can design a specialized system—e.g., a volatility- and volume-focused approach—simply by piping in the relevant series.

Configuration Guide

Custom Source Settings

You can configure up to 10 custom sources, each with identical configuration options. The settings below apply uniformly to all custom sources, allowing you to build a comprehensive feature set for your analysis.

Custom Source 1-10

Default: `close`

PURPOSE

Select custom sources with a specific analytical objective in mind to maximize the effectiveness of nearest-neighbor analysis.

IMPACT

Focused, thematic feature selection helps the model identify more meaningful patterns within your chosen domain of analysis.

RECOMMENDATIONS

- For volume analysis: combine volume, OBV, VWAP, and volume-weighted metrics
- For volatility: group ATR, standard deviation, and price range indicators
- For momentum: combine RSI, ROC, and other velocity-based metrics
- For price action: use various price levels and their derivatives

💡 **CONFIGURATION STRATEGY**

When configuring multiple custom sources:

1. Start with your most reliable data source in Custom Source 1
2. Add complementary sources in subsequent slots (2-10)
3. Avoid highly correlated sources across slots
4. Monitor performance impact as you add more sources

Normalization Options

Normalize Source

Default: `false`

PURPOSE

Standardizes input data to a common scale, ensuring fair comparison between different types of data.

IMPACT

Prevents any single source from dominating the model's decisions due to differences in scale.

RECOMMENDATIONS

- Enable when combining sources with different numerical ranges
- Particularly important for mixing price-based and oscillator data
- Consider leaving disabled for sources that should maintain their natural scale
- Use consistently across related sources for balanced analysis

> ⓘ **OPTIMIZATION TIP**
>
> Use normalization to ensure all features contribute equally to the model's decisions, preventing any single feature from dominating the analysis.

Example Use Cases

WaveTrend 3D Integration

- **Scenario**: You have a multi-timeframe WaveTrend 3D script that outputs three oscillators, each for a different timeframe (e.g., 12-hour, 1-day, 2-day).
- **Implementation**: Select each oscillator from the Custom Source drop-downs. Optionally, enable normalization to align them with price-based features.

Volume-Weighted Confluence

- **Scenario**: Your trading relies heavily on volume spikes and a custom "Volume Heatmap" script.
- **Implementation**: Load the volume script on the chart, pick the relevant output fields (heatmap, on-balance volume, or custom volume ratio) as a custom series, and normalize if necessary.

Multi-Factor System

- **Scenario**: You want to combine multiple types of market analysis by incorporating both momentum and volatility factors.
- **Implementation**:
    - Use "Custom Source 1" for a momentum indicator (like RSI or MACD)

  - Use "Custom Source 2" for a volatility metric (like ATR or Bollinger Band Width)
  - The Lorentzian Classification model can then identify patterns where both momentum and volatility conditions align for potential trade setups

Implementation Tips

Setting Up Custom Sources

- **Pro Tip**: Before assigning a slot, load the external indicator(s) you wish to reference on the chart. This ensures that the new data series appear in the drop-down list for selection.

Using Normalization

- **When to Enable**: Enable normalization when combining sources with widely different scales (e.g., volume in millions vs. oscillator from 0-100)
- **When to Disable**: Consider leaving normalization off for sources that should maintain their natural scale relationships

Key Takeaways

1. **Plan Your Inputs**: Before adding custom sources, clarify what you want to test or emphasize
2. **Normalize Wisely**: If your chosen series vary greatly in scale, enabling normalization maintains balance across features
3. **Experiment Responsibly**: More features can mean richer insights but also increased complexity
4. **Adapt Over Time**: As the market changes or new data sources arise, revisit your custom inputs

> **PERFORMANCE NOTE**
>
> Since custom sources reference data series that are already pre-computed in other indicators on your chart, there's very minimal computational overhead when using them. You're essentially just piping in existing calculations, making custom sources more efficient than built-in sources. Feel free to experiment with as many custom sources as you'd like—they won't incur the same performance impact as adding additional built-in sources.

Example Configurations

Multi-Timeframe Analysis Setup:

- **Custom Source 1:** WaveTrend 3D (12-hour timeframe, Normalize: Enabled) - *Reasoning: Captures slower, higher timeframe momentum.*

- **Custom Source 2:** WaveTrend 3D (1-day timeframe, Normalize: Enabled) - *Reasoning: Provides intermediate timeframe momentum context.*
- **Custom Source 3:** WaveTrend 3D (2-day timeframe, Normalize: Enabled) - *Reasoning: Offers broader, long-term momentum perspective.*
- **All other custom sources:** Disabled - *Reasoning: Focuses exclusively on multi-timeframe WaveTrend 3D data for a time-synchronized analysis. This configuration is tailored for users who want to perform multi-timeframe analysis using WaveTrend 3D. By integrating three different timeframes of WaveTrend 3D as custom sources, the model can capture oscillations and momentum shifts across multiple time horizons, providing a more comprehensive view of market dynamics. Normalization is enabled to ensure that the different WaveTrend 3D oscillators contribute equally to the model, despite potential scale differences.*

Volume-Focused Strategy:

- **Custom Source 1:** Volume Heatmap (Heatmap Output, Normalize: Enabled) - *Reasoning: Captures localized volume concentrations and order flow dynamics.*
- **Custom Source 2:** On Balance Volume (Normalize: Enabled) - *Reasoning: Measures cumulative buying and selling pressure over time.*
- **All other custom sources:** Disabled - *Reasoning: Concentrates solely on volume-based features to emphasize volume confirmation in signals. This configuration is designed for users whose strategies heavily rely on volume analysis. By using Volume Heatmap and On Balance Volume as custom sources, the model is optimized to identify patterns where volume confirms price movements. Volume Heatmap provides insights into localized volume concentrations, while On Balance Volume offers a broader view of buying and selling pressure. Disabling other custom sources ensures that the model's predictions are primarily driven by volume-based features.*

Enhanced Momentum System:

- **Custom Source 1:** RSI (Normalize: Enabled) - *Reasoning: Standard RSI for baseline momentum measurement.*
- **Custom Source 2:** MACD (Normalize: Enabled) - *Reasoning: MACD for trend and momentum confirmation.*
- **Custom Source 3:** Stochastic Oscillator (Normalize: Enabled) - *Reasoning: Stochastic Oscillator for overbought/oversold conditions and short-term momentum.*
- **All other custom sources:** Disabled - *Reasoning: Focuses on a combination of momentum indicators for a comprehensive momentum-based strategy. This configuration is geared towards users who want to build a robust momentum-based trading strategy. By combining RSI, MACD, and Stochastic Oscillator as custom sources, the model benefits from multiple perspectives on momentum. RSI provides a general measure of momentum, MACD offers trend*

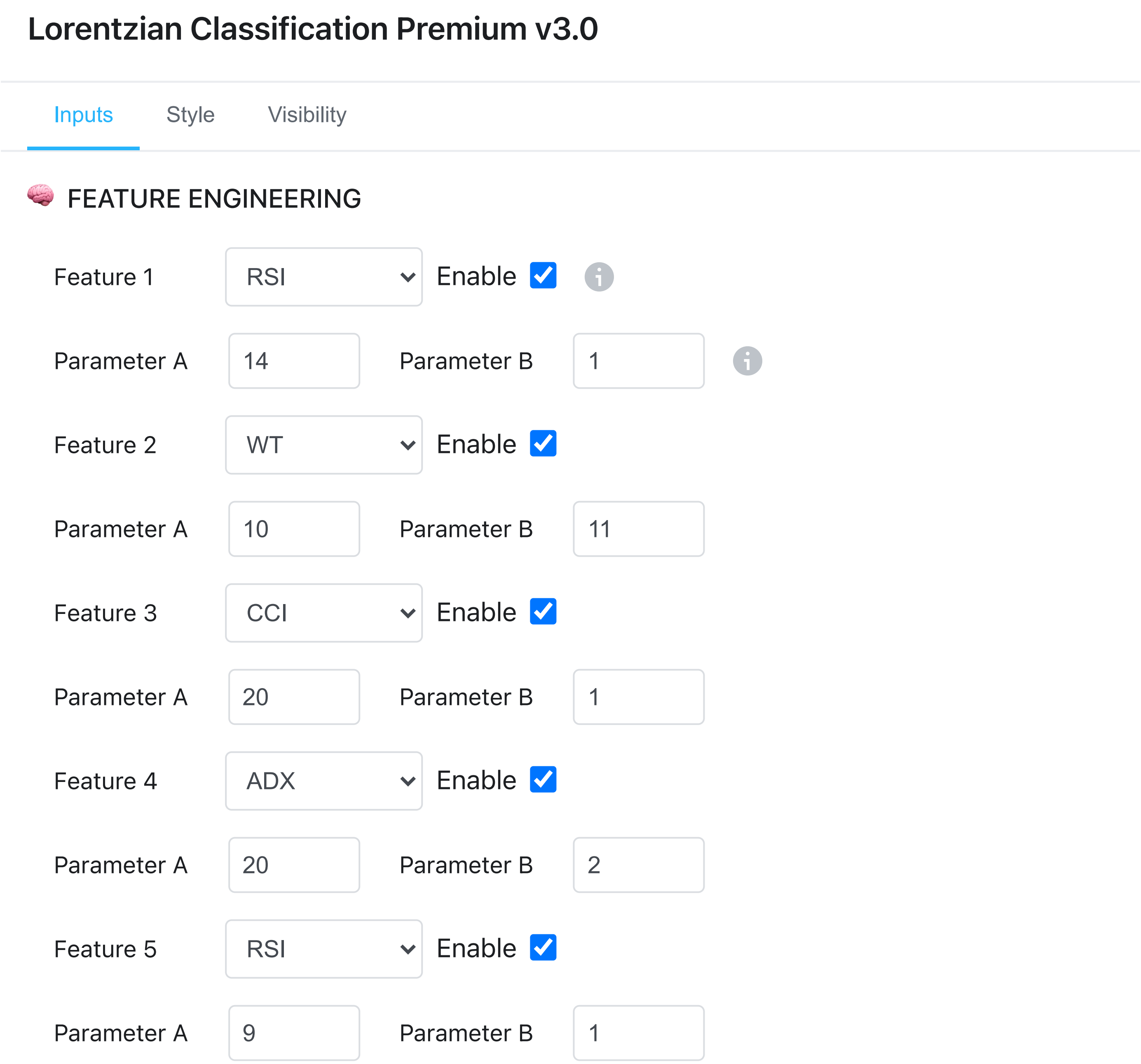
and momentum confirmation, and Stochastic Oscillator captures short-term momentum shifts and overbought/oversold conditions. Normalization ensures that these diverse momentum indicators contribute equally to the model's predictions.

Feature Engineering

Feature Engineering in the **Lorentzian Classification** indicator involves selecting, transforming, and curating meaningful data inputs before they feed into the machine learning model. Well-chosen features help the model recognize complex market patterns, reduce noise, and ultimately produce more robust trading signals. Conversely, poorly chosen or overly redundant features can lead to overfitting, slow computations, or misleading results.

Interactive Settings Explorer

Below is an interactive exploration of the Feature Engineering interface. Click any setting to instantly view its detailed documentation, default value, and purpose:



Lorentzian Classification Premium v3.0

🧠 FEATURE ENGINEERING

Feature 1	RSI	Enable ☑		
Parameter A	14	Parameter B	1	
Feature 2	WT	Enable ☑		
Parameter A	10	Parameter B	11	
Feature 3	CCI	Enable ☑		
Parameter A	20	Parameter B	1	
Feature 4	ADX	Enable ☑		
Parameter A	20	Parameter B	2	
Feature 5	RSI	Enable ☑		
Parameter A	9	Parameter B	1	

Feature 6	Custom 1	Enable ☐	
Parameter A	14	Parameter B	1
Feature 7	Custom 2	Enable ☐	
Parameter A	14	Parameter B	1
Feature 8	Custom 3	Enable ☐	
Parameter A	14	Parameter B	1
Feature 9	Custom 4	Enable ☐	
Parameter A	14	Parameter B	1
Feature 10	Custom 5	Enable ☐	
Parameter A	14	Parameter B	1
Feature 11	Custom 6	Enable ☐	
Parameter A	14	Parameter B	1
Feature 12	Custom 7	Enable ☐	
Parameter A	14	Parameter B	1
Feature 13	Custom 8	Enable ☐	
Parameter A	14	Parameter B	1
Feature 14	Custom 9	Enable ☐	
Parameter A	14	Parameter B	1
Feature 15	Custom 10	Enable ☐	
Parameter A	14	Parameter B	1

Why Feature Engineering Matters

Enhanced Predictive Power Financial data can be noisy and non-stationary. Meaningful transformations often reveal underlying signals the model might otherwise miss. For instance, using a short-period RSI plus a long-period RSI can help the model capture both short-term and longer-term momentum shifts.

Customized Analysis Your trading domain knowledge is invaluable. If you suspect that volume spikes matter more in certain regimes—or that mean reversion patterns are better detected by certain oscillators—feature engineering lets you align your inputs with your intuition.

Noise Reduction & Stability Smoothing raw data, filtering out outliers, or using rolling statistics can prevent the model from overreacting to random fluctuations. This is especially important when combining multiple indicators in PineScript.

Iterative Exploration There is rarely a "one-size-fits-all" feature set. The markets evolve, and so might your trading strategy. Feature engineering provides a flexible framework to adapt, add, or remove features as you gather new evidence or change market focus.

Multi-Method Strategies Combining diverse indicator types—momentum, trend, volatility, mean reversion—often leads to more robust modeling. For example, pairing RSI (momentum) with a WaveTrend oscillator (smoothed momentum) can give a balanced view of market oscillations.

Example Use Cases

1. Dual-Length RSI for Multiple Market Regimes

Scenario: You want one RSI tuned to short-term swings and another capturing broader trends. **Implementation**:

- Enable "RSI" in two feature slots with different lengths (e.g., RSI(9) and RSI(14)).
- Compare signals from short RSI to detect choppy market turnarounds, and longer RSI to capture more extended trends.

2. Momentum + Trend-Strength Combination

Scenario: You suspect that a momentum oscillator (like WaveTrend) plus a trend-strength measure (like ADX) will identify stronger breakouts. **Implementation**:

- Assign WaveTrend to Feature 1 and ADX to Feature 2 in the Lorentzian Classification UI.
- Ensure both are normalized so that ADX's higher scale doesn't overshadow WaveTrend values.

- Look for synergy: WaveTrend spotting momentum changes while ADX confirms if the move has real trend strength.

3. Mean Reversion + Volatility Filter

Scenario: You primarily trade mean reversions but only when volatility is above a certain threshold. **Implementation**:

- Use something like CCI or Bollinger-based data for your primary “mean reversion” feature.
- Add a volatility measure (such as ATR or a custom-coded volatility indicator) to Feature 2.
- Enable normalization for both features. If volatility is low, the model can down-rank those data points in neighbor matching.

4. Multi-Timeframe Signals

Scenario: You scalp on a 5-minute chart but want confirmation from a daily trend. **Implementation**:

- Load a daily ADX or daily RSI as a custom source in Feature 4 or 5.
- Combine it with your intraday oscillator in Features 1–3.
- Check if the higher-timeframe trend dimension helps filter out false positives on the lower timeframe.

5. Custom Oscillator + Doubling-Up for Emphasis

Scenario: You have a proprietary oscillator that strongly correlates with your market, and you want it to weigh more heavily in the classification. **Implementation**:

- Map the proprietary oscillator as a custom feature.
- Add the same feature again (with identical parameters) in another feature slot to double its influence.
- Confirm both features are normalized to ensure balanced impact among the rest of the indicators.

The Iterative Approach

Start Small Begin with one or two well-known indicators (e.g., RSI, WaveTrend). Verify that the Lorentzian Classification indicator produces stable, interpretable signals.

Add Features Gradually Introduce a single new feature—like CCI, ADX, or a custom metric—at a time. Observe changes in the model’s on-chart signals and the “Trade Stats” section. Overhauling your feature set all at once makes it hard to pinpoint what’s helping or hurting.

Check Normalization If a newly added feature has a different range (for example, ADX can grow larger than RSI), make sure the “Normalization” option is enabled to avoid skewing the distance calculations. Otherwise, that feature might dominate or dilute the influence of other features.

Evaluate Correlations Export your feature data to CSV and analyze correlations in Python or any statistical tool. High correlation between features can be redundant. On the other hand, features with low correlation might add unique information.

Backtest & Forward Test Although Lorentzian Classification includes on-chart trade stats, it is **not** a substitute for a full TradingView backtest or a more advanced backtesting framework.

- Use the Strategy Tester or a “Backtest Adapter” to get a realistic performance overview.
- Look for consistent improvements in metrics you care about (e.g., fewer false signals, higher win rate, better risk-adjusted returns).

Document Everything Jot down which features and parameter settings you’ve tried—along with how the signals changed. Feature engineering is often trial-and-error. Keeping records prevents repeated mistakes and accelerates learning.

Key Settings & Tips

Number of Features (Feature Count) Increasing the number of features (e.g., from 2 to 5 or more) can capture more complexity—but can also increase the risk of overfitting or PineScript timeouts. Many traders find a sweet spot around 5–8 features, **but** always test performance thoroughly.

Parameter A & Parameter B Many indicators have adjustable look-back periods or smoothing parameters. Small changes in these can drastically affect model predictions. For example, try RSI(14) vs. RSI(9) to capture different market regimes.

“Double-Up” for Weighting If you suspect one feature is more crucial than others, you can simulate giving it extra “weight” by using that feature multiple times with the same parameters. This approach can boost the feature’s influence in the distance calculation.

Normalization Checkboxes Each feature can be normalized by checking the relevant box in the UI. This ensures all features contribute on a level playing field. Not normalizing a large-range feature (e.g., ADX) can inadvertently dominate the model’s decisions.

Custom Features The indicator allows you to load external data streams or proprietary indicators. Make sure they produce valid floating-point values (not booleans or strings). If the external data is unbounded (like some volatility metrics), consider enabling normalization.

Performance & Timeouts If computations become slow or you get PineScript timeouts, try reducing the number of features or the maximum bars back. Another trick is **downsampling**: skipping bars or training on fewer historical bars to speed up calculations.

Pitfalls to Avoid

Feature Overload More features do **not** always yield better results. Beyond a certain point, "curse of dimensionality" can degrade accuracy and slow performance. Keep your feature set purposeful.

Unbalanced Scales Failing to normalize can allow high-range indicators (like ADX or unbounded CCI) to overshadow others.

Redundant Indicators Two highly correlated indicators (e.g., RSI and Stoch RSI with the same length) might not add unique value.

Boolean or Categorical Series Lorentzian Classification typically assumes float inputs. Passing a series of `1` and `0` can distort distance calculations. If you must encode a boolean, consider converting it to a fraction or smoothed float.

Key Takeaways

Align transforms with your hypothesis.
Track performance rigorously.
Refine continually.
Know when to stop.

Default: `undefined`

PURPOSE

Example Configurations

High-Precision Trend Following Configuration:

- **Number of Features:** 5
- **Feature 1:** RSI (Parameter A: 14, Parameter B: 1) - *Reasoning: Standard RSI for momentum.*
- **Feature 2:** WT (Parameter A: 10, Parameter B: 11) - *Reasoning: WaveTrend for smoothed momentum.*

- **Feature 3:** ADX (Parameter A: 20, Parameter B: 2) - *Reasoning: ADX for trend strength confirmation.*
- **Feature 4:** CCI (Parameter A: 20, Parameter B: 1) - *Reasoning: CCI for volatility-based momentum.*
- **Feature 5:** RSI (Parameter A: 9, Parameter B: 1) - *Reasoning: Short-period RSI for capturing short-term swings, complementing longer-period RSI.*
- **Normalization:** Enabled for all features - *Reasoning: Ensures balanced contribution from all features despite different scales. This configuration is designed for users who prioritize high-precision trend following. It combines a diverse set of momentum, trend strength, and volatility indicators to create a robust feature set. The dual RSI feature allows the model to capture both short-term and long-term momentum, while ADX and CCI add trend and volatility context. Normalization is crucial to ensure that all features contribute equally to the model's predictions, preventing any single indicator from dominating the analysis.*

Custom Volume-Based Strategy Configuration:

- **Number of Features:** 3
- **Feature 1:** Custom 1 (Volume Weighted MACD, Normalize: Enabled, Parameter A: 12, Parameter B: 26) - *Reasoning: Volume-weighted MACD to capture momentum confirmed by volume.*
- **Feature 2:** Custom 2 (On Balance Volume, Normalize: Enabled, Parameter A: 14, Parameter B: 1) - *Reasoning: On Balance Volume to measure buying and selling pressure.*
- **Feature 3:** ADX (Parameter A: 20, Parameter B: 2) - *Reasoning: ADX for trend strength confirmation, ensuring volume breakouts are trend-supported.*
- **All other features:** Disabled - *Reasoning: Focuses the model on volume and trend strength, avoiding dilution from other feature types. This configuration is tailored for users who believe volume is a critical factor in market movements. It integrates volume-based indicators with trend strength to identify volume-confirmed breakouts and avoid false signals. Volume Weighted MACD and On Balance Volume provide complementary perspectives on volume and momentum, while ADX ensures that trades are aligned with strong trends. Normalization is enabled for custom sources to balance their scale with ADX.*

Low-Latency Scalping Configuration:

- **Number of Features:** 3
- **Feature 1:** RSI (Parameter A: 7, Parameter B: 1) - *Reasoning: Short-period RSI for quick momentum detection in scalping.*
- **Feature 2:** WT (Parameter A: 5, Parameter B: 5) - *Reasoning: Short-period WaveTrend for fast, smoothed momentum signals.*

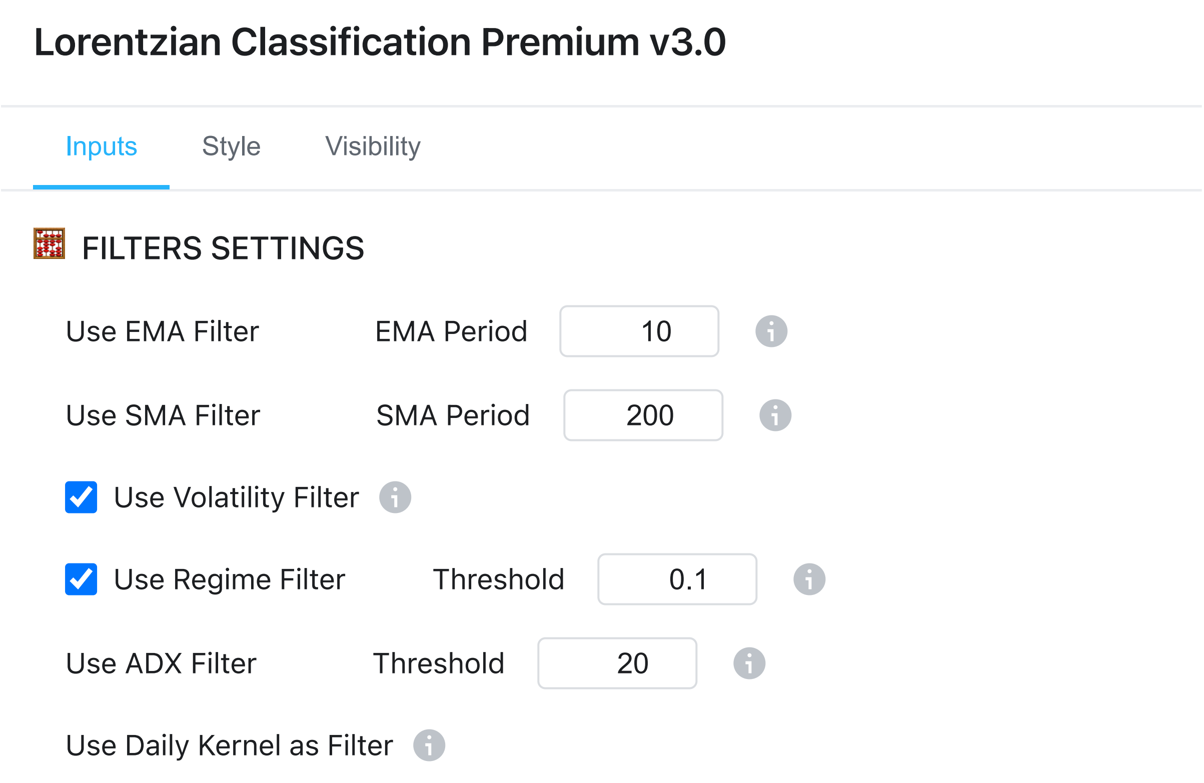
- **Feature 3:** CCI (Parameter A: 10, Parameter B: 1) - *Reasoning: Short-period CCI for capturing short-term volatility swings.*
- **Max Bars Back:** 1500 - *Reasoning: Reduces computational load and emphasizes recent price action relevant to scalping.*
- **Normalization:** Enabled for all features - *Reasoning: Maintains balanced feature contribution despite different indicator scales. This configuration is optimized for scalping strategies that require low latency and quick responses. It uses shorter lookback periods for RSI, WT, and CCI to increase responsiveness to intraday price movements. Reducing 'Max Bars Back' further minimizes computational load, ensuring the indicator remains fast and reactive. Normalization is enabled to maintain balanced feature contribution. This setup prioritizes speed and responsiveness over long-term trend analysis, making it suitable for fast-paced intraday trading.*

Filters Settings

These settings control the filtering mechanisms of the Lorentzian Classification model. They help refine signals by applying various technical filters like moving averages, volatility conditions, and trend strength measurements.

Interactive Settings Explorer

Below is an interactive exploration of the Filter Settings interface. Click any setting to instantly view its detailed documentation, default value, and purpose - a quick alternative to scrolling through the full reference below:



Lorentzian Classification Premium v3.0

FILTERS SETTINGS

Use EMA Filter EMA Period 10

Use SMA Filter SMA Period 200

☑ Use Volatility Filter

☑ Use Regime Filter Threshold 0.1

Use ADX Filter Threshold 20

Use Daily Kernel as Filter

Detailed Settings Reference

Quick Reference Table

Filter	Default	Purpose	Key Parameter (Range)
Volatility Filter	True	Avoid quiet markets	ATR-based logic

Filter	Default	Purpose	Key Parameter (Range)
Regime Filter	True	Trend vs. range detection	Threshold (-0.5–0.5)
ADX Filter	False	Trend strength confirmation	ADX (15–25)
EMA Filter	False	Align with short-term momentum	Period (20–100)
SMA Filter	False	Long-term directional bias	Period (50–200)
Daily Kernel	False	Higher timeframe confluence	Uses daily trend

Moving Average Filters

EMA Filter

Default: `false`

PURPOSE
Applies an EMA-based trend filter to signals, ignoring trades outside the EMA direction.

IMPACT
Filters out trades that conflict with short-term trend direction.

RECOMMENDATIONS

- Enable for trend-following strategies
- Combine with volatility filter
- Adjust EMA period to match your timeframe

SMA Filter

Default: `false`

PURPOSE
Applies an SMA-based filter for higher-level trend validation.

IMPACT
Filters out trades that conflict with a slower, more stable trend measure.

RECOMMENDATIONS

- Use for swing trading or longer timeframes

- Combine with fast EMA or ADX filters
- Consider 200 SMA for major trend identification

Market Condition Filters

Volatility Filter

Default: `true`

PURPOSE

Avoids trading in low-volatility conditions that can produce choppy signals.

IMPACT

Helps skip periods where price moves are weak or range-bound.

RECOMMENDATIONS

- Keep enabled for most strategies
- Especially useful in momentum trading
- Disable in steady, range-bound markets if you prefer mean reversion

> **TECHNICAL DETAIL**
>
> The volatility filter helps avoid choppy market conditions where signals may be less reliable. It's especially useful in ranging markets.

Regime Filter

Default: `true`

PURPOSE

Distinguishes trending vs. ranging environments using a threshold-based logic.

IMPACT

Prevents applying trend-following logic in sideways conditions.

RECOMMENDATIONS

- Crucial if your strategy performs poorly in chop
- Adjust threshold for faster or slower detection
- Start with 0.1 and adjust based on market

Trend Strength Filters

ADX Filter

Default: `false`

PURPOSE
Checks ADX to ensure sufficient trend strength before entering trades.

IMPACT
Skipping trades when ADX is below threshold can reduce false breakouts.

RECOMMENDATIONS

- Enable if you only want strong trends
- Combine with smaller threshold if your market is often volatile
- Consider 20-25 range for standard trend detection

Multi-timeframe Filters

Daily Kernel Filter

Default: `false`

PURPOSE
Uses daily timeframe kernel logic to confirm signals on lower timeframes.

IMPACT
Ensures you only trade in harmony with big-picture daily trends or expansions.

RECOMMENDATIONS

- Enable for multi-timeframe strategies
- Useful for avoiding counter-trend trades
- Consider computational impact on your platform

Filter Combinations

Understanding how different filters work together is key to optimizing your configuration:

- **EMA + Volatility**: Great for trending markets
- **Regime + ADX**: Excellent for avoiding choppy conditions
- **Daily Kernel + SMA**: Strong multi-timeframe confluence
- **Volatility + ADX**: Powerful for breakout validation

> ⓘ **WHY THESE DEFAULTS?**
>
> Default values are chosen to offer a good balance between signal quality and quantity. They provide meaningful filtering while avoiding over-restriction of trading opportunities.

Summary

The filter settings provide multiple layers of signal refinement for the Lorentzian Classification model. While the defaults offer a solid starting point, understanding each filter's purpose allows you to create a custom configuration that matches your trading style and market conditions.

Try experimenting with the interactive demo above to see how different filter combinations affect signal generation. Each setting has detailed documentation that appears when you click on it in the interface.

Example Configurations

Conservative Trend Following:

- **Volatility Filter:** True - *Reasoning: Avoids trading in choppy, low-volatility environments, reducing the risk of false signals.*
- **Regime Filter:** True (Threshold: 0.1) - *Reasoning: Confirms a trending market before entering trades, ensuring the model is operating in a favorable regime.*
- **ADX Filter:** True (Threshold: 25) - *Reasoning: Ensures sufficient trend strength for reliable signals, filtering out weak or uncertain trends.*
- **EMA Filter:** True (Period: 50) - *Reasoning: Aligns trades with short-term momentum, providing additional confirmation of the trend.*
- **SMA Filter:** True (Period: 200) - *Reasoning: Validates trades with the long-term trend, ensuring alignment with the overall market direction.*
- **Daily Kernel Filter:** False - *Reasoning: Focuses on the current timeframe without multi-timeframe confirmation, simplifying the analysis. This configuration is designed for Fixed Income/Conservative Traders who prioritize safety and trend alignment. It employs multiple filters to ensure that only high-probability, trend-aligned signals are generated, minimizing risk and false positives. The SMA and EMA filters align with long and short term trends, ADX ensures trend strength, and the Volatility and Regime filters avoid choppy markets. This setup is ideal for long-term investors or those with a low-risk tolerance.*

Aggressive Scalping:

- **Volatility Filter:** False - *Reasoning: Allows trading in all market conditions, including low-volatility periods, to maximize signal frequency.*
- **Regime Filter:** True (Threshold: -0.2) - *Reasoning: Captures even weak trends, suitable for scalping, where small price movements can be profitable.*
- **ADX Filter:** False - *Reasoning: Disables trend strength confirmation for faster signal generation, allowing for quicker entries and exits.*
- **EMA Filter:** False - *Reasoning: Removes short-term trend alignment for quicker entries, prioritizing speed over confirmation.*
- **SMA Filter:** False - *Reasoning: Disables long-term trend validation for faster exits, essential for capturing small profits.*
- **Daily Kernel Filter:** False - *Reasoning: Focuses solely on the current timeframe for scalping, ignoring higher timeframe context. This configuration is tailored for Alert Automation Users and Performance Tweakers focused on scalping strategies. It disables most filters to maximize signal frequency and responsiveness, prioritizing speed over signal quality. The Regime filter with a negative threshold is used to capture even weak trends, while the absence of other filters allows for quick entries and exits in fast-moving markets. This setup is best suited for experienced scalpers who can manage the increased risk of false signals and require a high volume of trades.*

Multi-Timeframe Confluence:

- **Volatility Filter:** True - *Reasoning: Avoids trading in choppy, low-volatility environments, improving signal reliability.*
- **Regime Filter:** True (Threshold: 0.1) - *Reasoning: Confirms a trending market before entering trades, ensuring the model operates in a favorable regime.*
- **ADX Filter:** False - *Reasoning: Allows for signals even if the trend strength is not high on the current timeframe, providing more opportunities for entry.*
- **EMA Filter:** False - *Reasoning: Avoids over-filtering and maintains responsiveness on lower timeframes, allowing for quicker reactions to price movements.*
- **SMA Filter:** False - *Reasoning: Avoids over-filtering and maintains responsiveness on lower timeframes, prioritizing short-term opportunities.*
- **Daily Kernel Filter:** True - *Reasoning: Ensures trades align with the higher timeframe (Daily) trend, providing a broader perspective on market direction. This configuration is designed for Visual Confluence Traders who seek confirmation from a higher timeframe before entering trades. It enables the Daily Kernel Filter to ensure that all signals align with the daily trend, providing a broader perspective on market direction. The Volatility and Regime filters help to avoid trading in choppy or ranging markets, while the SMA and EMA filters are disabled to maintain responsiveness on lower timeframes. This setup is ideal for traders who use*

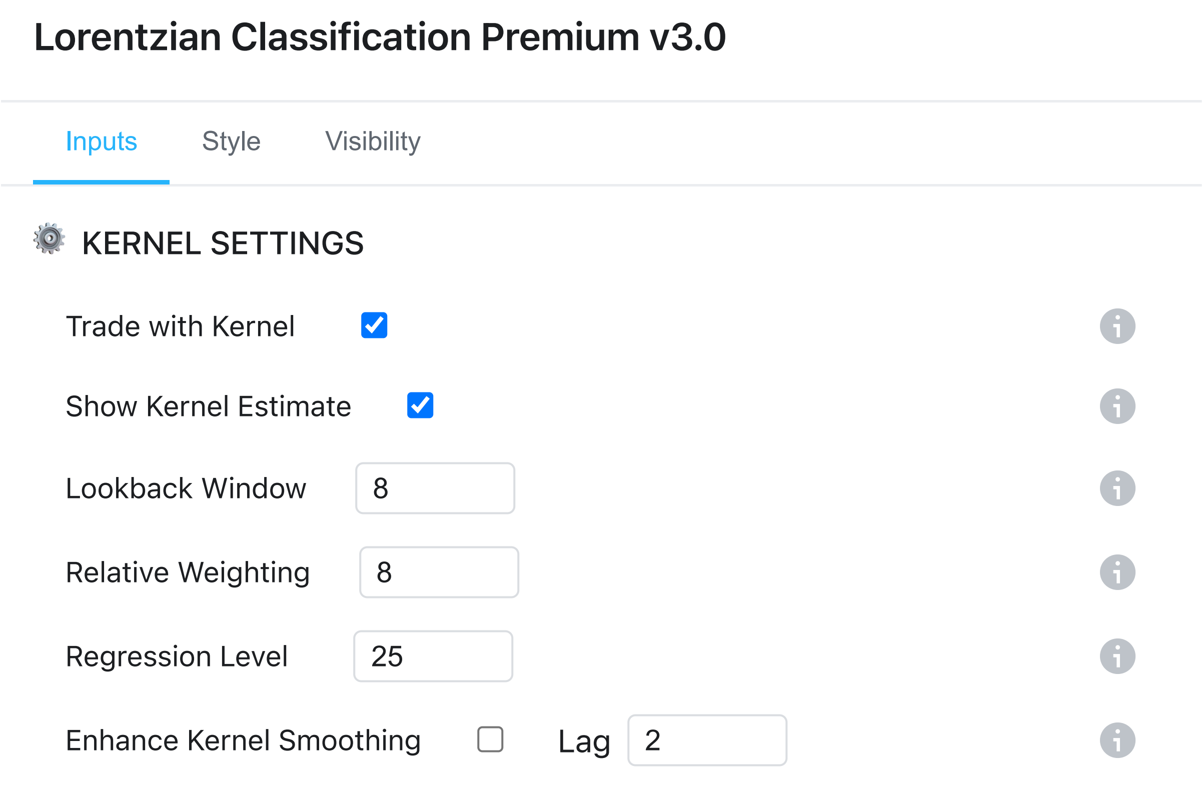
Lorentzian Classification as a confluence tool to confirm their trading decisions based on higher timeframe analysis.

Understanding Kernel Settings

Advanced parameters that control how kernel regression estimates the trend and adapts to market fluctuations. Kernel functions estimate the underlying trend by weighting past price data non-linearly, offering more flexibility than standard moving averages.

Interactive Settings Explorer

Below is an interactive exploration of the Kernel Settings interface. Click any setting to instantly view its detailed documentation, default value, and purpose:



Lorentzian Classification Premium v3.0

KERNEL SETTINGS

Trade with Kernel ☑

Show Kernel Estimate ☑

Lookback Window 8

Relative Weighting 8

Regression Level 25

Enhance Kernel Smoothing □ Lag 2

Core Concepts

Why Use Kernel Settings?

- **Adaptive Smoothing**: More nuanced than an SMA/EMA, capturing subtle directional shifts.
- **Noise Reduction**: Helps filter out random price spikes or dips.

Quick Reference Table

Setting	Default	Purpose	Range (Recommended)
Trade with Kernel	True	Activates kernel-based trend	On/Off
Show Kernel Estimate	True	Displays the kernel line	On/Off
Enhance Kernel Smooth	False	Reduces color jitter via crossovers	On/Off
Lookback Window	8	Bars used for kernel calc	3–50
Relative Weighting	8.0	Emphasis on recent vs. older data	0.25–25
Regression Level	25	Start bar for regression	2–25
Lag	2	Delay in crossover detection	1–2

Configuration Parameters

Core Settings

1. **Trade with Kernel**
   - Toggles kernel-based logic for signals instead of raw price or standard MAs.
   - Often more stable in choppy markets.
2. **Show Kernel Estimate**
   - Visual feedback on the chart for real-time trend interpretation.
3. **Enhance Kernel Smoothing**
   - Additional logic (crossovers/thresholds) to reduce frequent color flips.

Advanced Parameters

4. **Lookback Window**
   - Shorter = more reactive, risk of overfitting. Longer = smoother, but can lag.

5. **Relative Weighting**

    - High weighting emphasizes recent bars, capturing fresh market dynamics.
    - Low weighting spreads the influence over a broader time window.

6. **Regression Level**

    - Skips early bars that might distort baseline, focusing on more stable data from bar X onward.

7. **Lag**

    - Deliberate delay to confirm crossovers and reduce whipsaws.

Trading Configurations

Reactive Intraday

- Lookback: ~5
- Relative Weighting: 5–8
- Regression Level: 2–5
- Lag: 1

Steady Swing

- Lookback: 10–12
- Relative Weighting: ~8 (default)
- Regression Level: 20–25
- Lag: 2

Optimization Tips

Best Practices

- **Layer with Filters**: Combine kernel-based direction with ADX/Regime filters.
- **Conditional Adjustments**: In high volatility, shorten the window. In calmer periods, lengthen.
- **Iterate Regularly**: Document changes & observe improvements.

Key Takeaway: Fine-tuning kernel settings gives you powerful control over how quickly the model senses and reacts to directional changes, balancing responsiveness with noise reduction.

Example Configurations

Smooth Trend Following:

- **Trade with Kernel:** True - *Reasoning: Enables kernel-based trend estimation, providing a more adaptive approach than simple moving averages.*
- **Show Kernel Estimate:** True - *Reasoning: Provides visual confirmation of the trend, allowing for quick assessment of market direction.*
- **Enhance Kernel Smoothing:** True - *Reasoning: Reduces noise and provides a clearer trend signal, minimizing whipsaws and false signals.*
- **Lookback Window:** 12 - *Reasoning: Captures a broader range of price action for smoother trend estimation, reducing sensitivity to short-term fluctuations.*
- **Relative Weighting:** 5.0 - *Reasoning: Balances responsiveness and stability in trend detection, giving more weight to recent data while still considering historical context.*
- **Regression Level:** 20 - *Reasoning: Skips initial volatile bars for a more stable baseline, focusing on more established trends.*
- **Lag:** 2 - *Reasoning: Reduces whipsaws and confirms crossover signals, improving the reliability of trend entries. This configuration is designed for Backtest Optimizers and Algorithm Curious Learners interested in smooth trend estimation. It employs enhanced kernel smoothing and a longer lookback window to create a smoother, more stable kernel estimate, ideal for trend-following strategies and reducing noise. A higher Regression Level skips initial bars for a better fit, and the Lag setting helps to confirm crossover signals.*

Responsive Intraday Trading:

- **Trade with Kernel:** True - *Reasoning: Enables kernel-based trend estimation, providing a more adaptive approach than simple moving averages.*
- **Show Kernel Estimate:** True - *Reasoning: Provides visual confirmation of the trend, allowing for quick assessment of market direction.*
- **Enhance Kernel Smoothing:** False - *Reasoning: Allows for quicker reactions to price changes, essential for scalping.*
- **Lookback Window:** 5 - *Reasoning: Captures short-term price movements for scalping, increasing responsiveness to intraday fluctuations.*
- **Relative Weighting:** 10.0 - *Reasoning: Emphasizes recent price action for responsiveness, prioritizing the most recent data.*
- **Regression Level:** 5 - *Reasoning: Fits the kernel tighter to recent data, capturing short-term trends.*
- **Lag:** 1 - *Reasoning: Minimizes delay in crossover detection for timely entries, crucial for scalping. This configuration is designed for Performance Tweakers and Reactive Intraday*

Traders focused on responsiveness. It uses a shorter lookback window and disables kernel smoothing for a more reactive kernel estimate, suitable for intraday trading and capturing quick price movements. A lower Regression Level fits the kernel tighter to recent data, and a lower Lag minimizes delay in crossover detection.

Long-Term Trend Analysis:

- **Trade with Kernel:** True - *Reasoning: Enables kernel-based trend estimation, providing a more adaptive approach than simple moving averages.*
- **Show Kernel Estimate:** True - *Reasoning: Provides visual confirmation of the trend, allowing for quick assessment of market direction.*
- **Enhance Kernel Smoothing:** True - *Reasoning: Reduces noise and provides a clearer trend signal, minimizing whipsaws and false signals.*
- **Lookback Window:** 20 - *Reasoning: Captures a broader range of price action for smoother trend estimation, reducing sensitivity to short-term fluctuations.*
- **Relative Weighting:** 2.0 - *Reasoning: Emphasizes longer-term trends and reduces sensitivity to short-term fluctuations, providing a more stable view of the market.*
- **Regression Level:** 30 - *Reasoning: Further stabilizes the kernel by focusing on later, more stable data, reducing the impact of initial volatility.*
- **Lag:** 2 - *Reasoning: Reduces whipsaws and confirms crossover signals, improving the reliability of trend entries. This configuration is designed for Fixed Income/Conservative Traders and Algorithm Curious Learners interested in long-term trend analysis. It employs a longer lookback window and lower relative weighting to emphasize longer-term trends and reduce sensitivity to short-term fluctuations. A higher Regression Level further stabilizes the kernel by focusing on later, more stable data.*

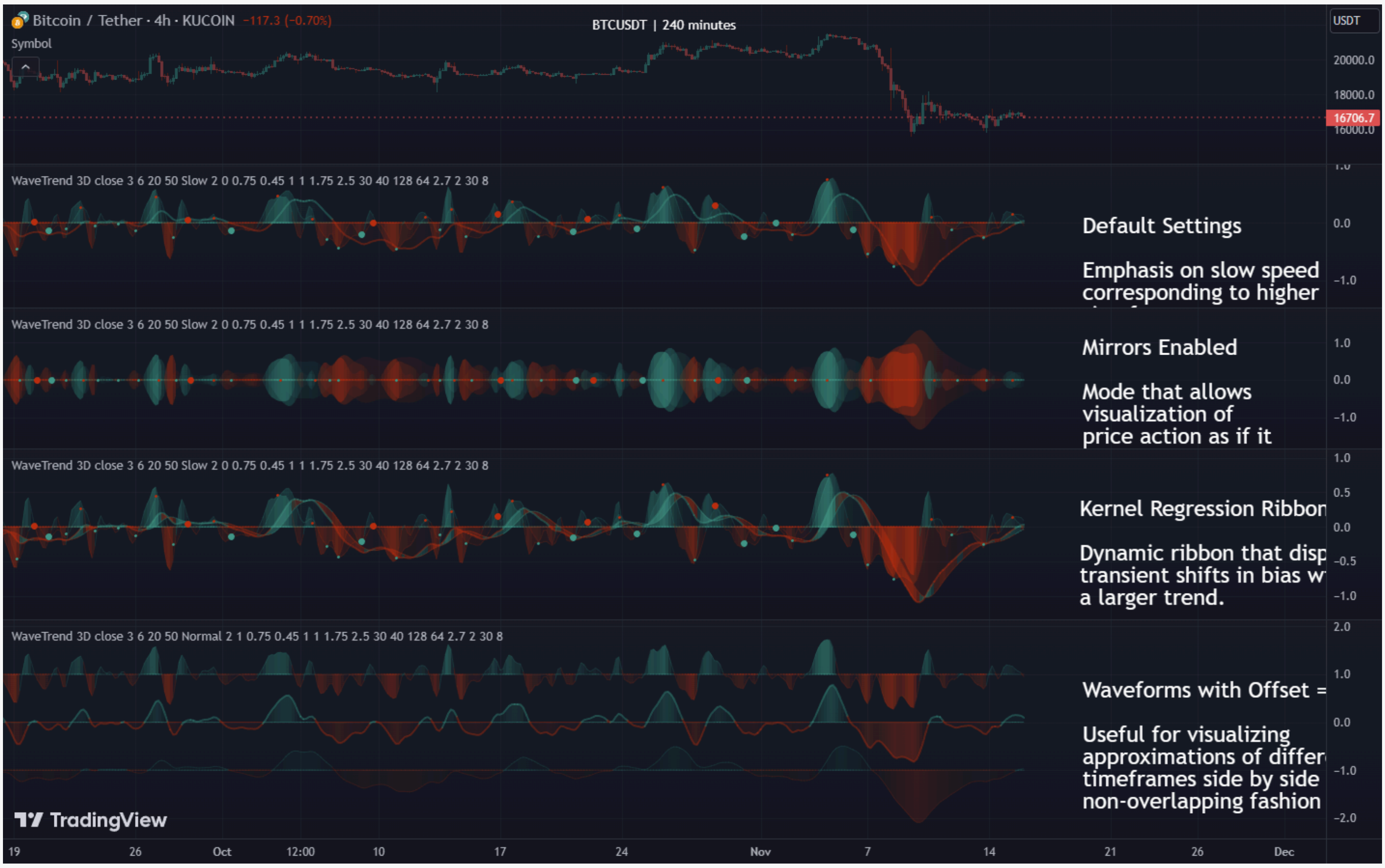
Wavetrend 3D

WHAT YOU'LL LEARN

This comprehensive guide will introduce you to the Wavetrend 3D indicator—a multi-timeframe, three-dimensional approach for analyzing market trends, inflection points, and potential trading opportunities. You'll learn why this enhanced version goes beyond the classic WaveTrend approach, and how its modern signal-processing techniques help provide deeper insight into market transitions.

Introduction

The **Wavetrend 3D** indicator is an advanced reimagining of the classic WaveTrend, integrating multiple timeframes (or "dimensions") into a single view. By default, most indicators analyze one timeframe at a time, which can miss crucial "bigger picture" context when the market transitions from ranging to trending states. Wavetrend 3D aims to close this gap by using advanced filtering (e.g., Butterworth filters) and hyperbolic tangent normalization to highlight market cycles and potential triggers across multiple frequencies.



Default Settings

Emphasis on slow speed corresponding to higher

Mirrors Enabled

Mode that allows visualization of price action as if it

Kernel Regression Ribbon

Dynamic ribbon that disp transient shifts in bias w a larger trend.

Waveforms with Offset =

Useful for visualizing approximations of differ timeframes side by side non-overlapping fashion

Fig 1: A high-level look at Wavetrend 3D's layered oscillators.

Why This Matters

Traditional WaveTrend indicators provide valuable signals based on overbought/oversold zones and divergences. However, they can become less reliable during sudden regime shifts. **Wavetrend 3D** tackles these weaknesses by:

- **Capturing multi-timeframe alignment:** Shows how smaller oscillations (e.g., 6-hour) interact with larger ones (e.g., daily).
- **Reducing whipsaw confusion:** Emphasizes trend continuity over quick reversals.
- **Offering new visual modes:** Such as mirrored mode, enabling traders to interpret bullish vs. bearish currents more intuitively.

ⓘ **HISTORICAL BACKGROUND**

WaveTrend Classic, made popular on platforms like TradingView, ThinkorSwim, and MetaTrader, originated from AIQ Systems in the 1980s under the name "Trading Channel Index (TCI)." Over time, it became a go-to tool for detecting overbought/oversold levels and divergences. **Wavetrend 3D** builds on this legacy with modern filtering and multi-timeframe synchronization.

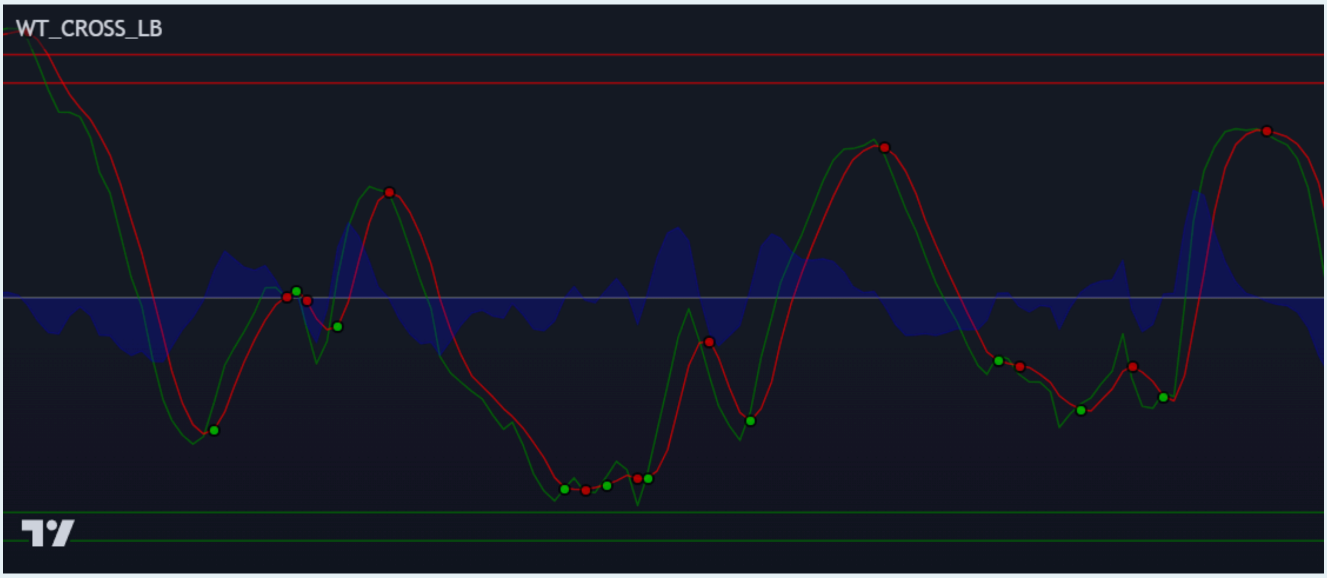


Fig 2: Original WaveTrend (Classic) approach, popularized since the 1980s.

How Wavetrend 3D Works

At its core, Wavetrend 3D applies modern signal-processing techniques:

1. **3D Visualization:** It plots a “fast,” “normal,” and “slow” oscillator. Each captures a different timeframe perspective, effectively compressing multi-timeframe analysis into one chart.

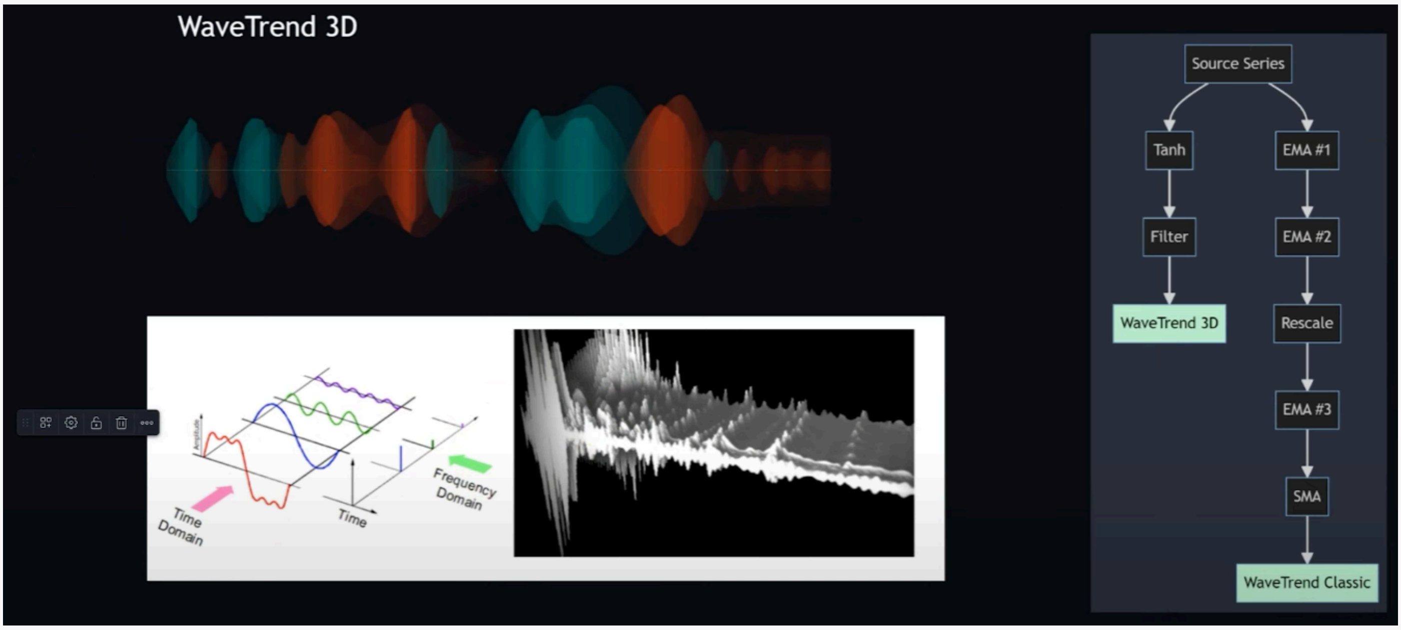


Fig 3: Conceptual visualization of multi-frequency oscillators in a "3D" wave.

2. **Noise Reduction:** A Butterworth filter and optional smoothing methods (e.g., EMA or CoG) reduce lag while retaining critical signal structure.
3. **Trigger Circles:** “Small circles” appear when the fast and normal oscillators cross, while “big circles” identify potential “trigger wave” divergences—often early signals of major price momentum.

What This Means for Traders

By viewing market data in three dimensions, traders can:

- **Spot multi-timeframe confluence:** Identify moments when short- and long-term oscillators align, often signaling stronger potential moves.
- **Track regime shifts:** Watch how the slow oscillator crosses its zero line, a key pivot that often implies a trend change.
- **Adapt to extremes:** Overbought/oversold boundaries tailored to Wavetrend 3D’s normalized scale offer more nuanced alerts.

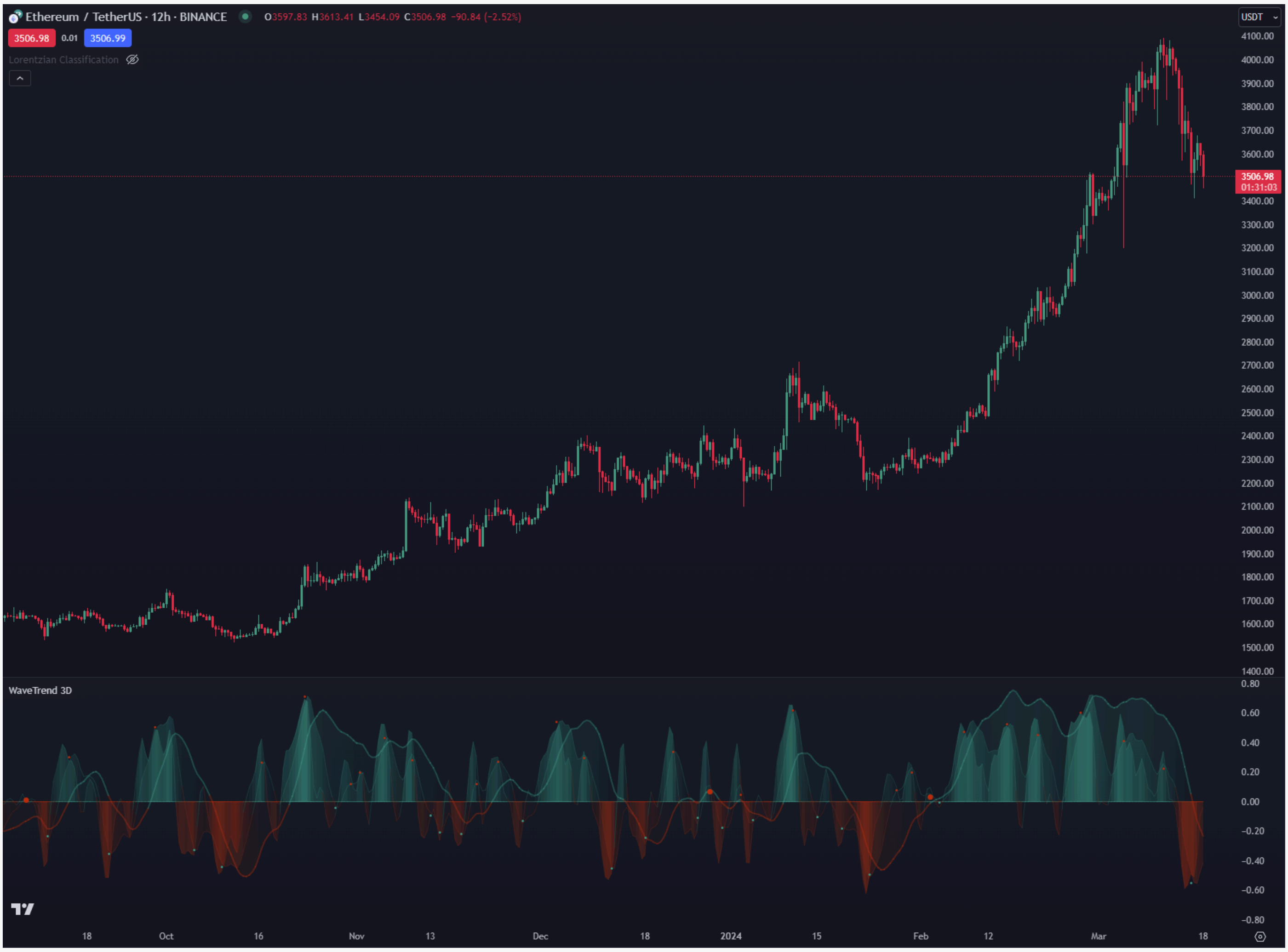


Fig 4: Wavetrend 3D overlaid on multiple timeframes for Ethereum (6h, 12h, 24h).

What This Guide Covers

In the pages that follow, you'll learn:

- **Core Concepts of Wavetrend 3D:** A deep dive into oscillators, small vs. big circle signals, mirrored mode, and smoothing options.
- **Practical Use Cases:** How to interpret circle signals, watch for divergences, and adapt settings like EMA or CoG.
- **Advanced Features:** Discover kernel estimators, zero-lag smoothing, and how to leverage overbought/oversold zones in a 3D context.

Understanding Core Concepts

While traditional WaveTrend indicators rely heavily on overbought/oversold conditions and single-timeframe divergences, **Wavetrend 3D** redefines the landscape by stitching together multiple “speeds” (or timeframes) into one harmonized view. Below, we break down key ideas powering Wavetrend 3D.

Multi-Timeframe Oscillators

Why “3D” Matters

Markets don’t move in purely 2D. Trend shifts on a 4-hour chart can overshadow signals on a 1-hour chart. **Wavetrend 3D** addresses this by blending three oscillators:

- **Fast** (e.g., ~6-hour on a 12-hour chart)
- **Normal** (the main timeframe, e.g., 12-hour)
- **Slow** (a broader perspective, e.g., daily)

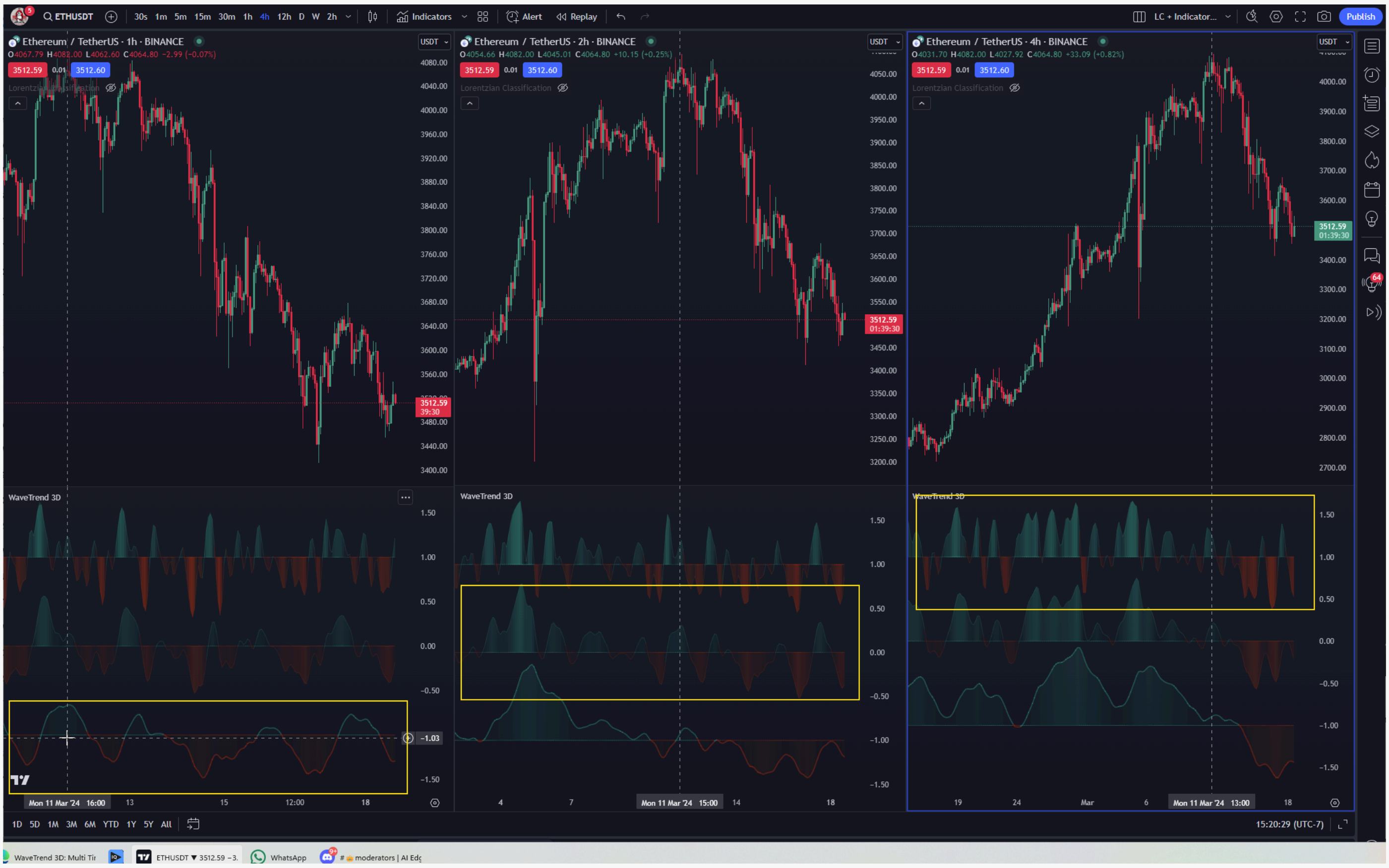

ETHUSDT
30s 1m 5m 15m 30m 1h 4h 12h D W 2h
Indicators
Alert
Replay
Publish
Ethereum / TetherUS · 1h · BINANCE
Ethereum / TetherUS · 2h · BINANCE
Ethereum / TetherUS · 4h · BINANCE
Lorentzian Classification
3512.59
3512.60
WaveTrend 3D
1.50
1.00
0.50
0.00
-0.50
-1.00
-1.50
Mon 11 Mar '24 16:00
Mon 11 Mar '24 15:00
Mon 11 Mar '24 13:00
1D 5D 1M 3M 6M YTD 1Y 5Y All
15:20:29 (UTC-7)

Fig 1: Separate fast, normal, and slow oscillators on different timeframes.

This reveals how multiple timeframes intersect, highlighting when **fast** and **normal** align or conflict with the **slow** oscillator.

Small Circles & Big Circles

Small Circles (Fast vs. Normal Crosses)

Small circles appear when the fast oscillator crosses the normal oscillator:

- **Red Circle Above Zero**: Potential bearish correction.
- **Green Circle Below Zero**: Potential bullish reversal.

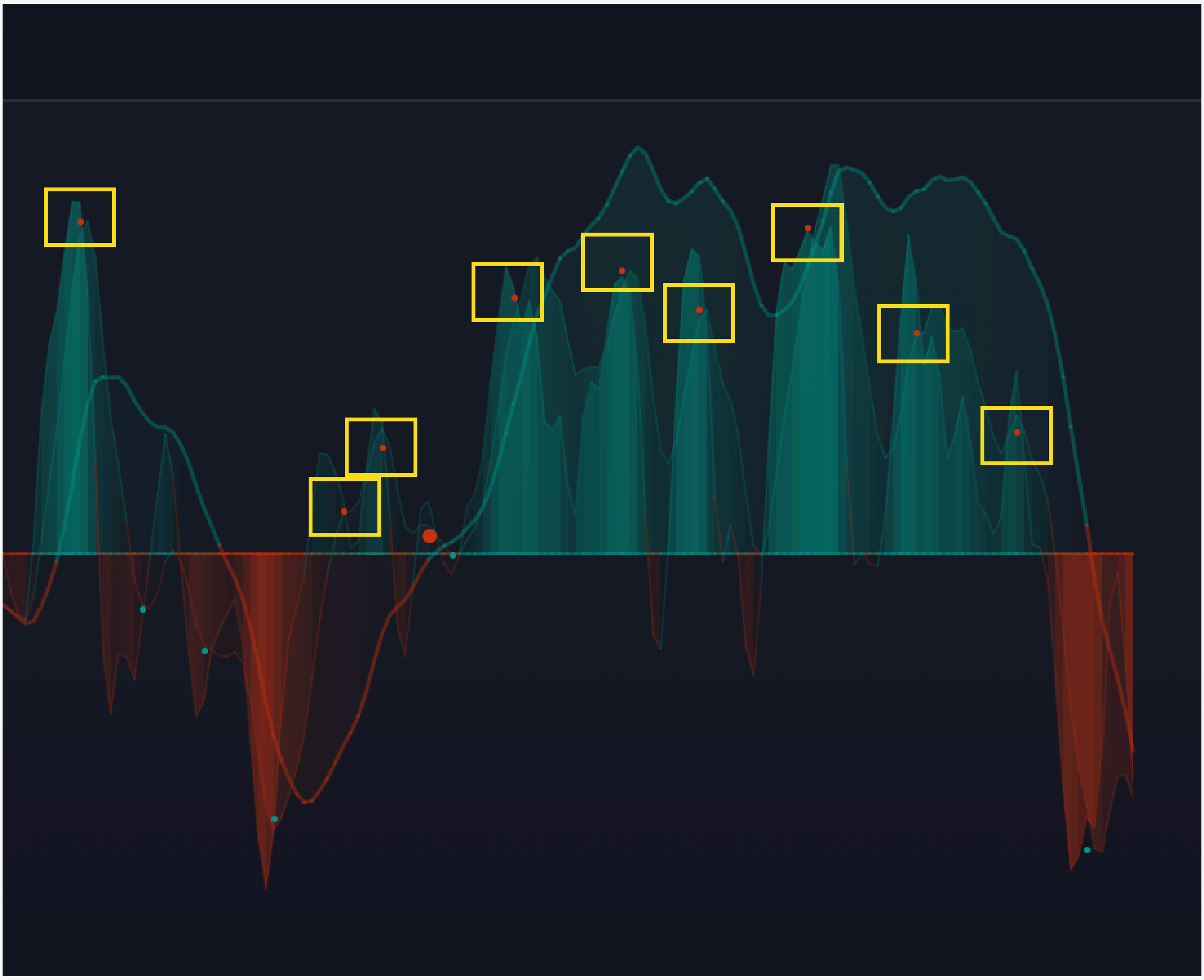


Fig 2: Fast crosses normal above zero, generating a small red circle signal.

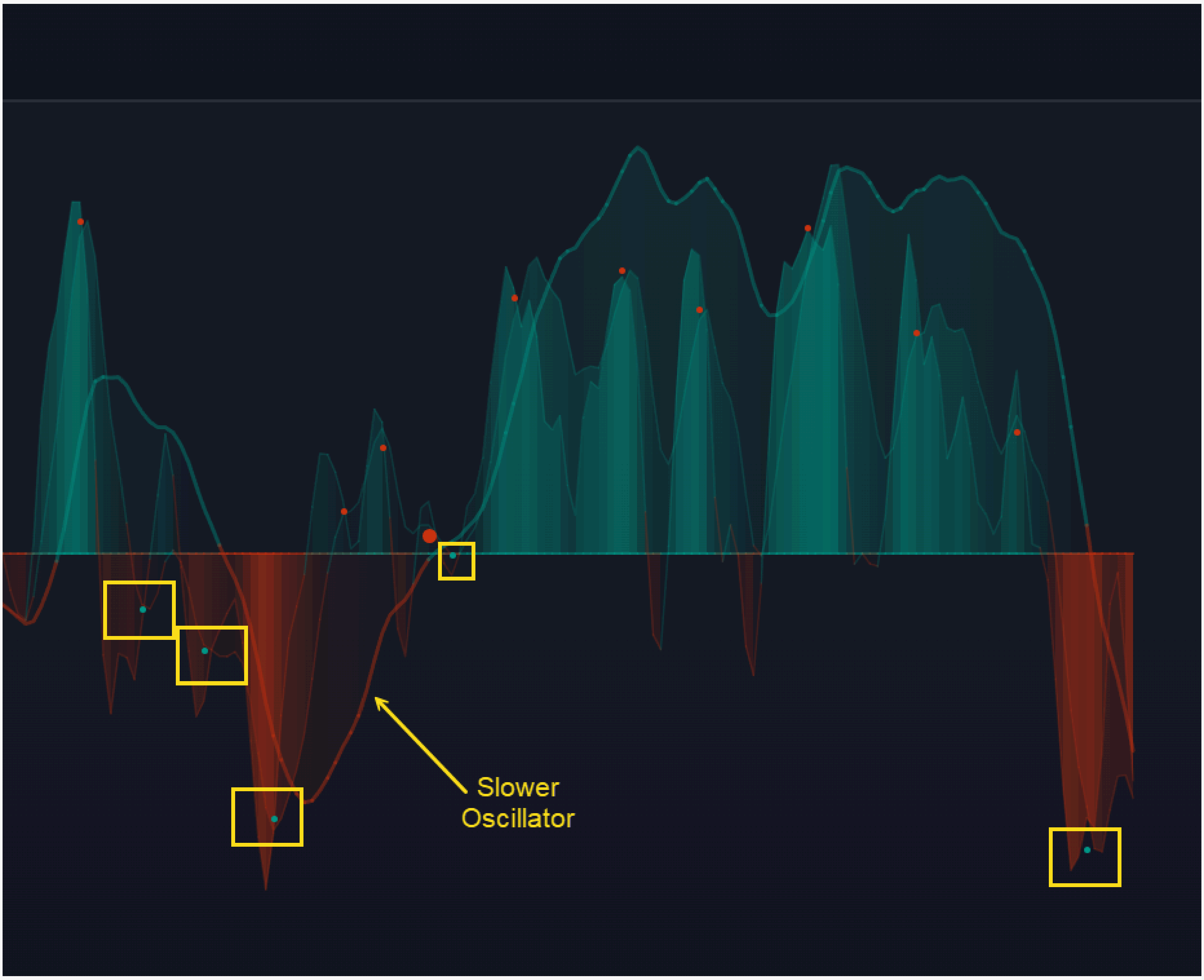


Fig 3: Fast crosses normal below zero, generating a small green circle signal.

Big Circles (Potential Trigger Waves)

Big circles mark a possible divergence wave—often a reduced-amplitude crossover compared to a prior wave. These can precede stronger moves.

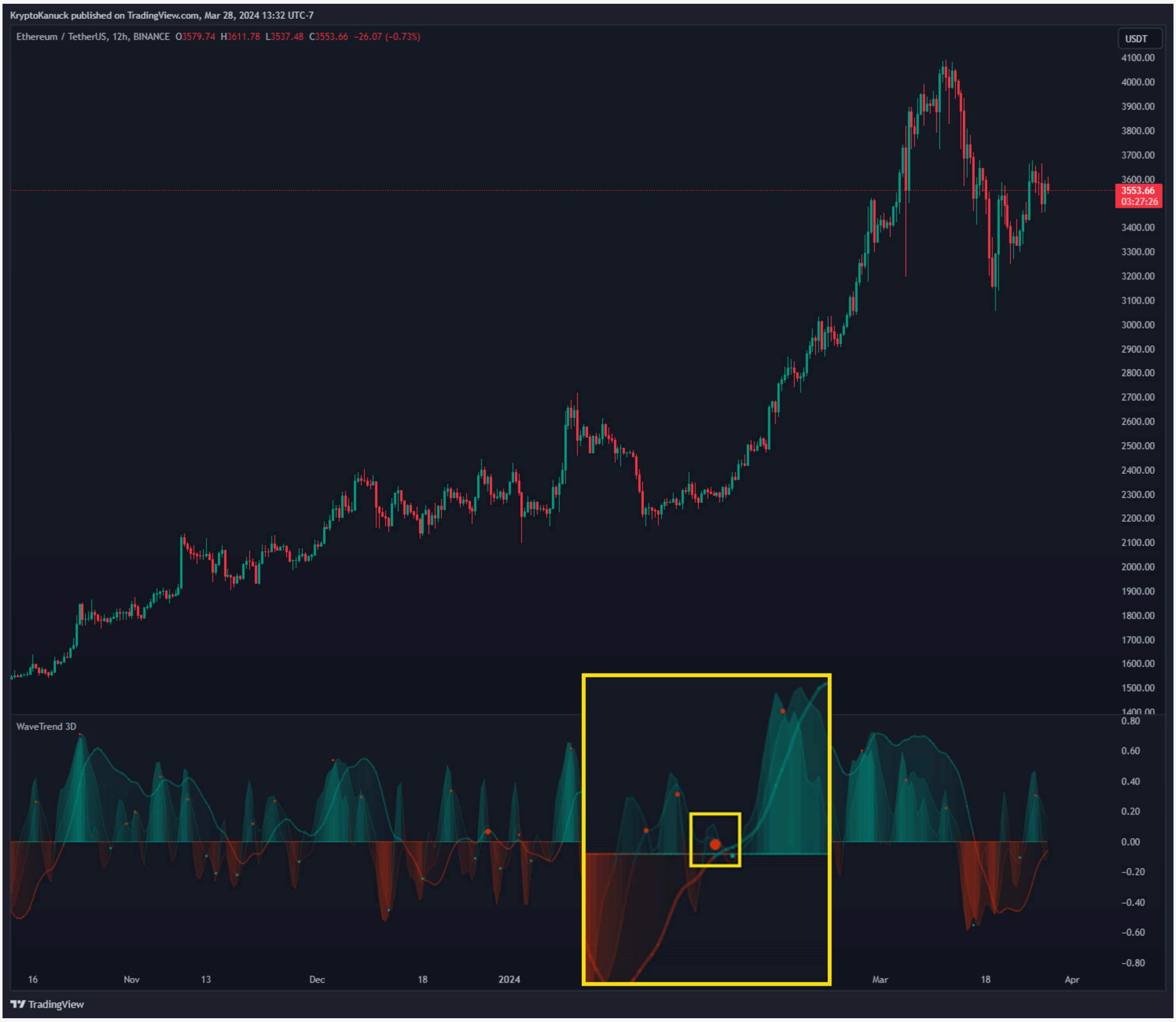
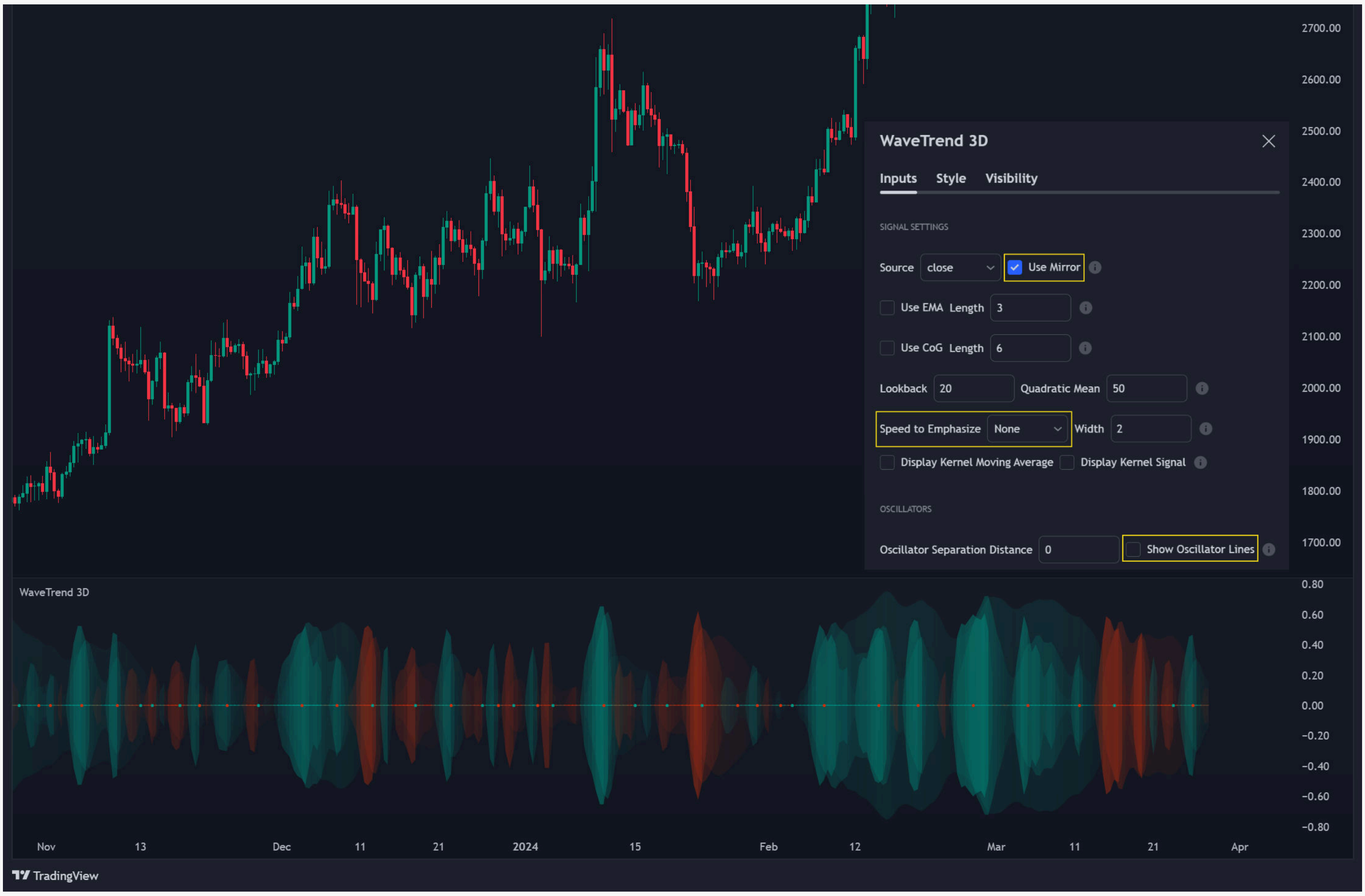


Fig 4: A bullish trigger wave forming at lower amplitude than a prior bearish wave.

Mirrored Mode

Mirroring signals across zero provides an alternate lens: bullish oscillations reflect upward, bearish oscillations invert downward.



WaveTrend 3D

Inputs Style Visibility

SIGNAL SETTINGS

Source close ☑ Use Mirror

□ Use EMA Length 3

□ Use CoG Length 6

Lookback 20 Quadratic Mean 50

Speed to Emphasize None Width 2

□ Display Kernel Moving Average □ Display Kernel Signal

OSCILLATORS

Oscillator Separation Distance 0 □ Show Oscillator Lines

Fig 5: Mirrored mode for a different perspective on oscillator alignment.

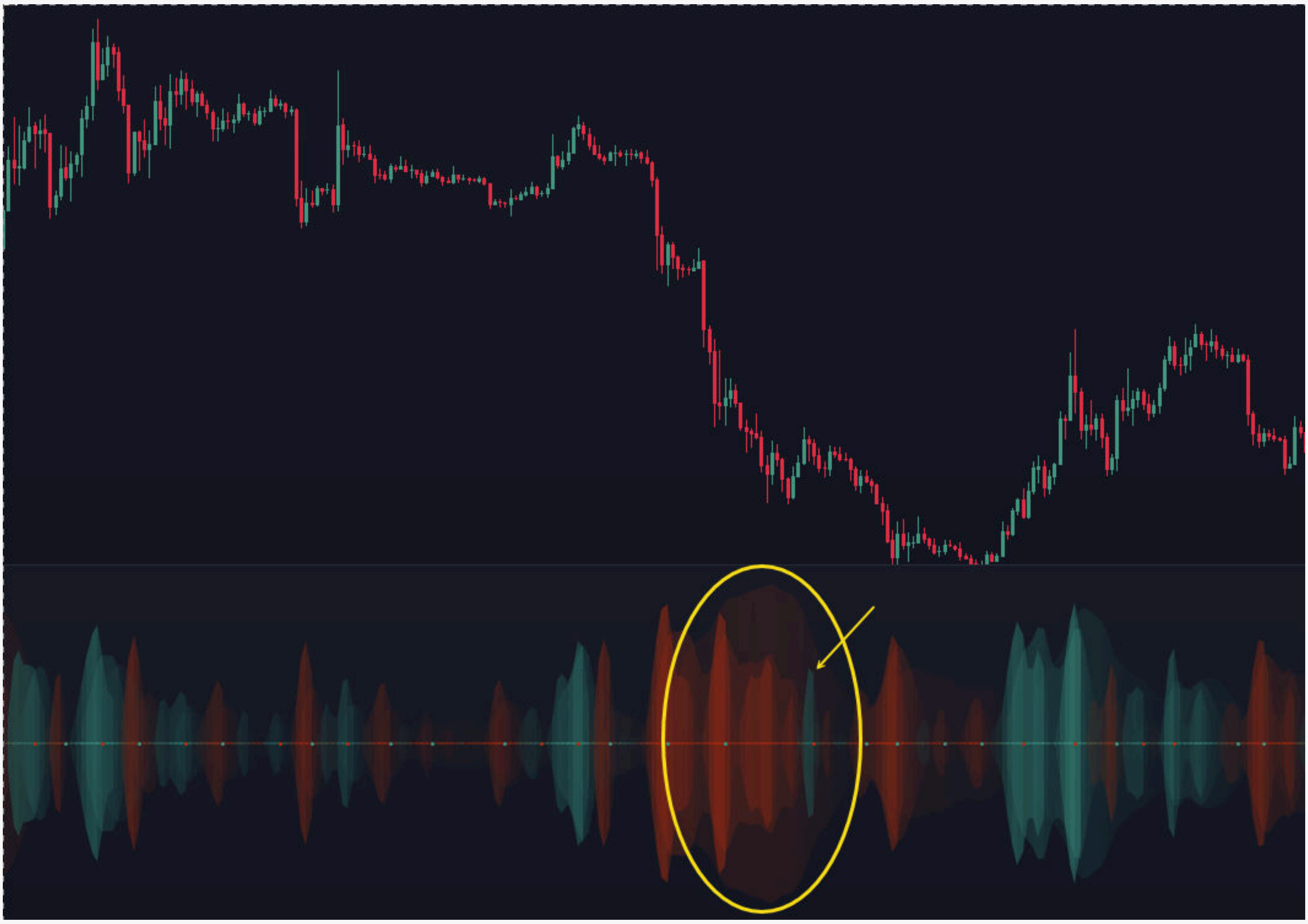


Fig 6: Identifying smaller bullish/bearish "bubbles" within a larger wave.

Divergence detection can also become clearer in mirrored space.

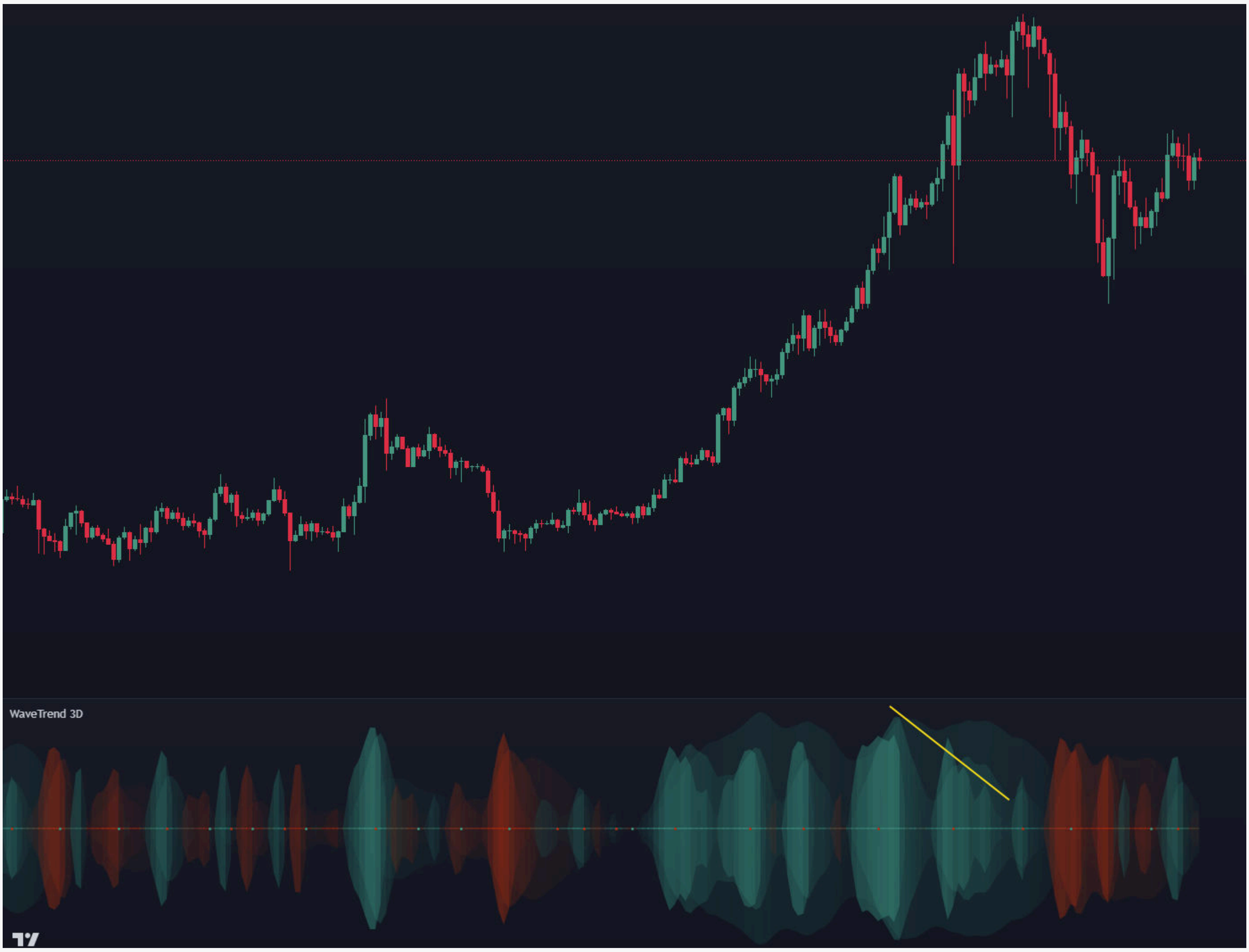
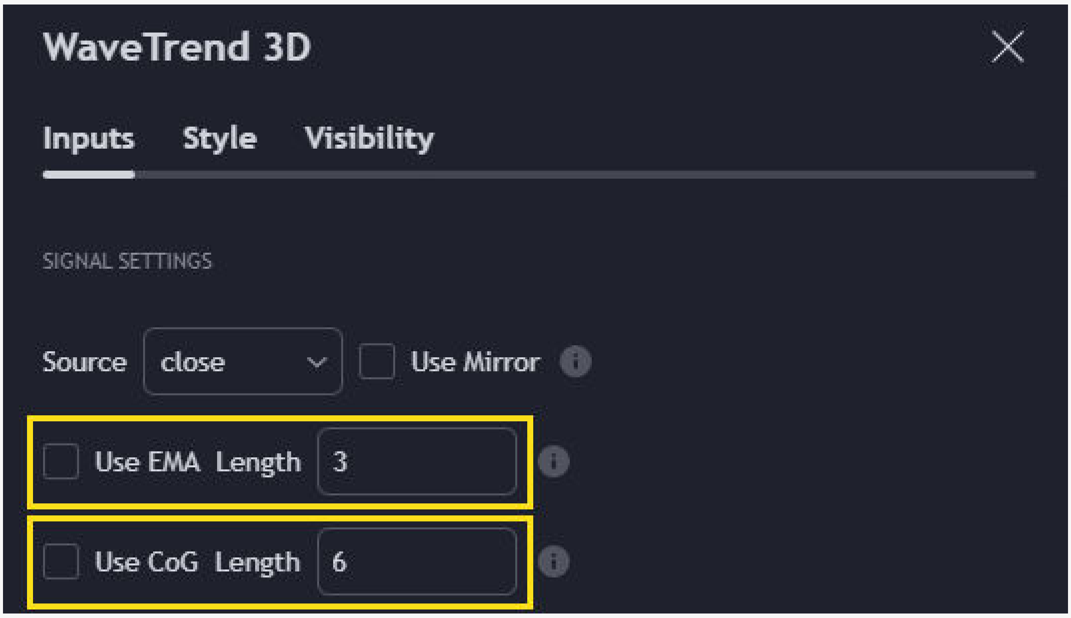


Fig 7: A smaller mirrored wave relative to its predecessor often signals an upcoming reversal.

EMA & CoG Smoothing

Exponential Moving Average (EMA)

By default, Wavetrend 3D has optional EMA smoothing. Increasing the **EMA Length** offers stronger noise reduction, at the cost of slight delay.



WaveTrend 3D

SIGNAL SETTINGS

Source close □ Use Mirror

□ Use EMA Length 3

□ Use CoG Length 6

Fig 8: Adjusting EMA length to smooth the oscillators further.

Center of Gravity (CoG)

CoG is a zero-lag, adaptive smoothing method. Its drastic look can often highlight inflection points earlier.

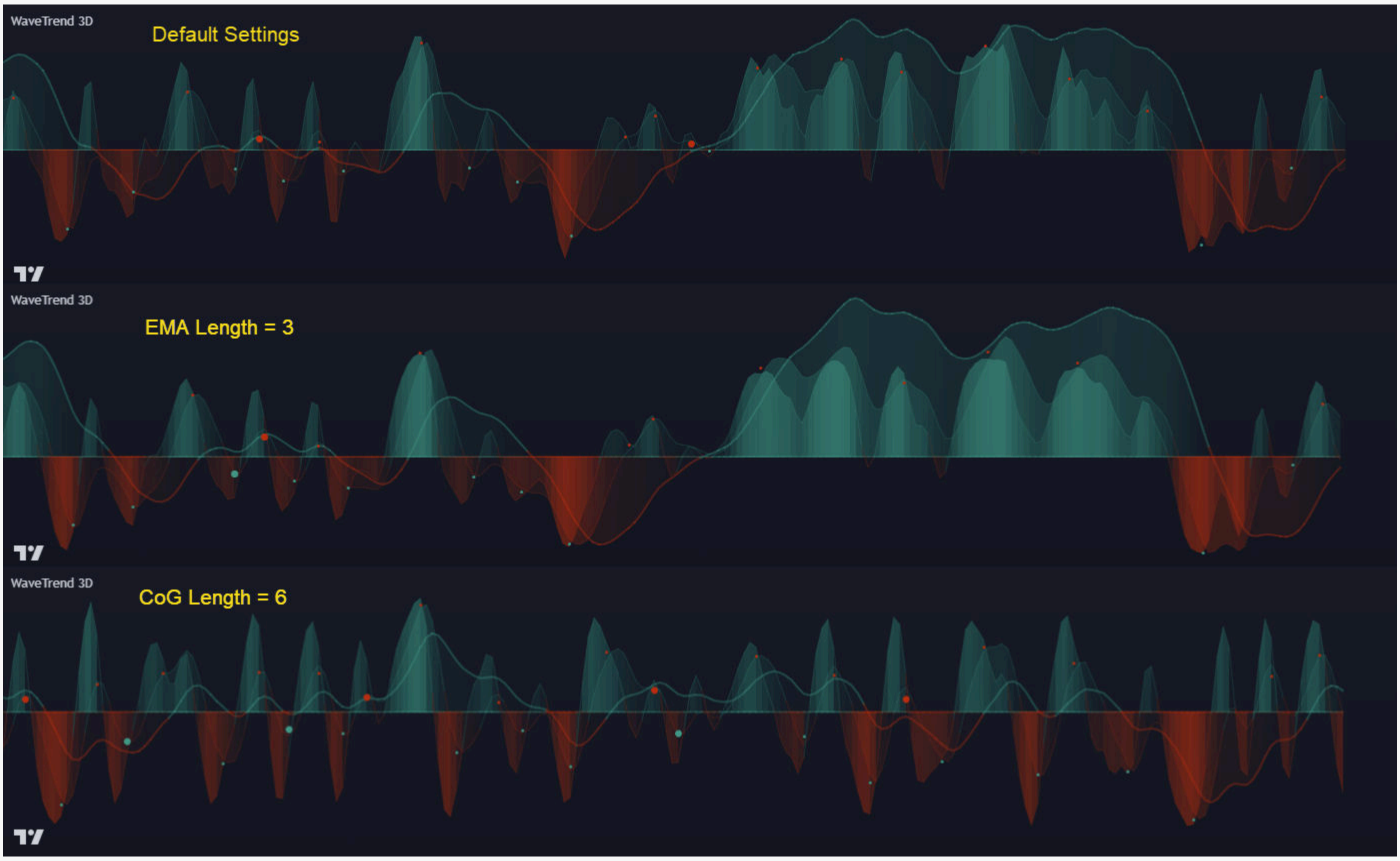


Fig 9: CoG smoothing giving a different waveform shape compared to EMA.

Speed to Emphasize

Select which oscillator (fast, normal, or slow) to emphasize. You can bold it or adjust the width for visual clarity.

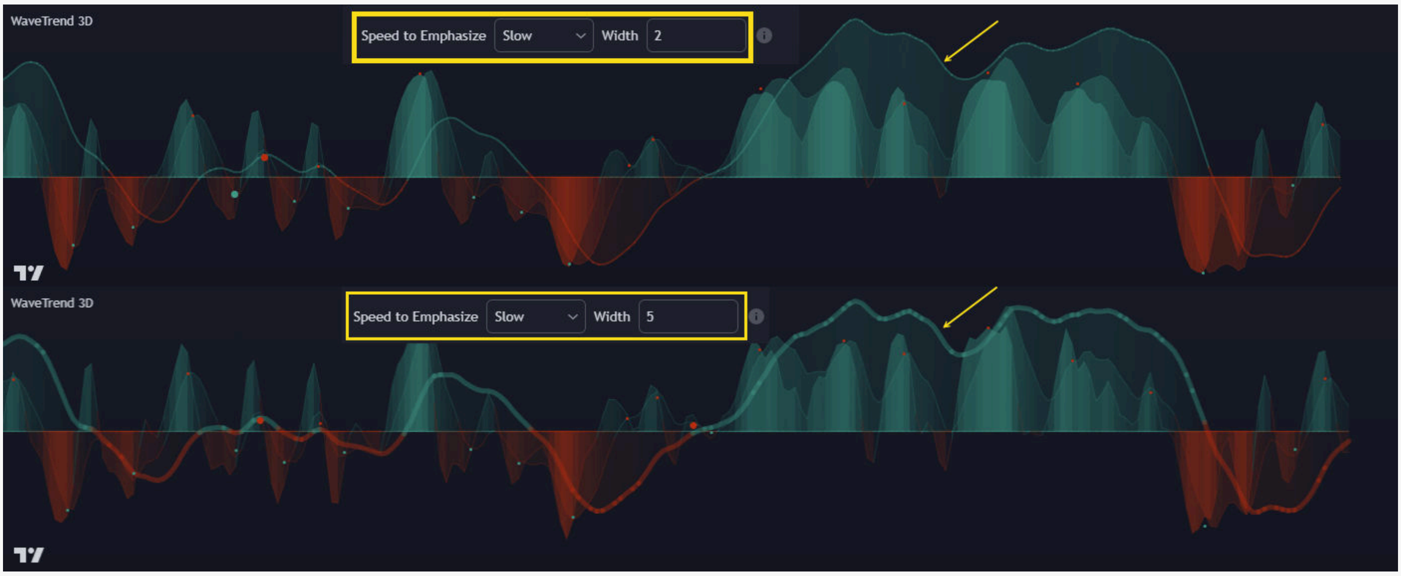


Fig 10: Emphasizing the slow oscillator to spot major regime shifts.

Crossing the zero line on the slow oscillator can signal a strong trend change.

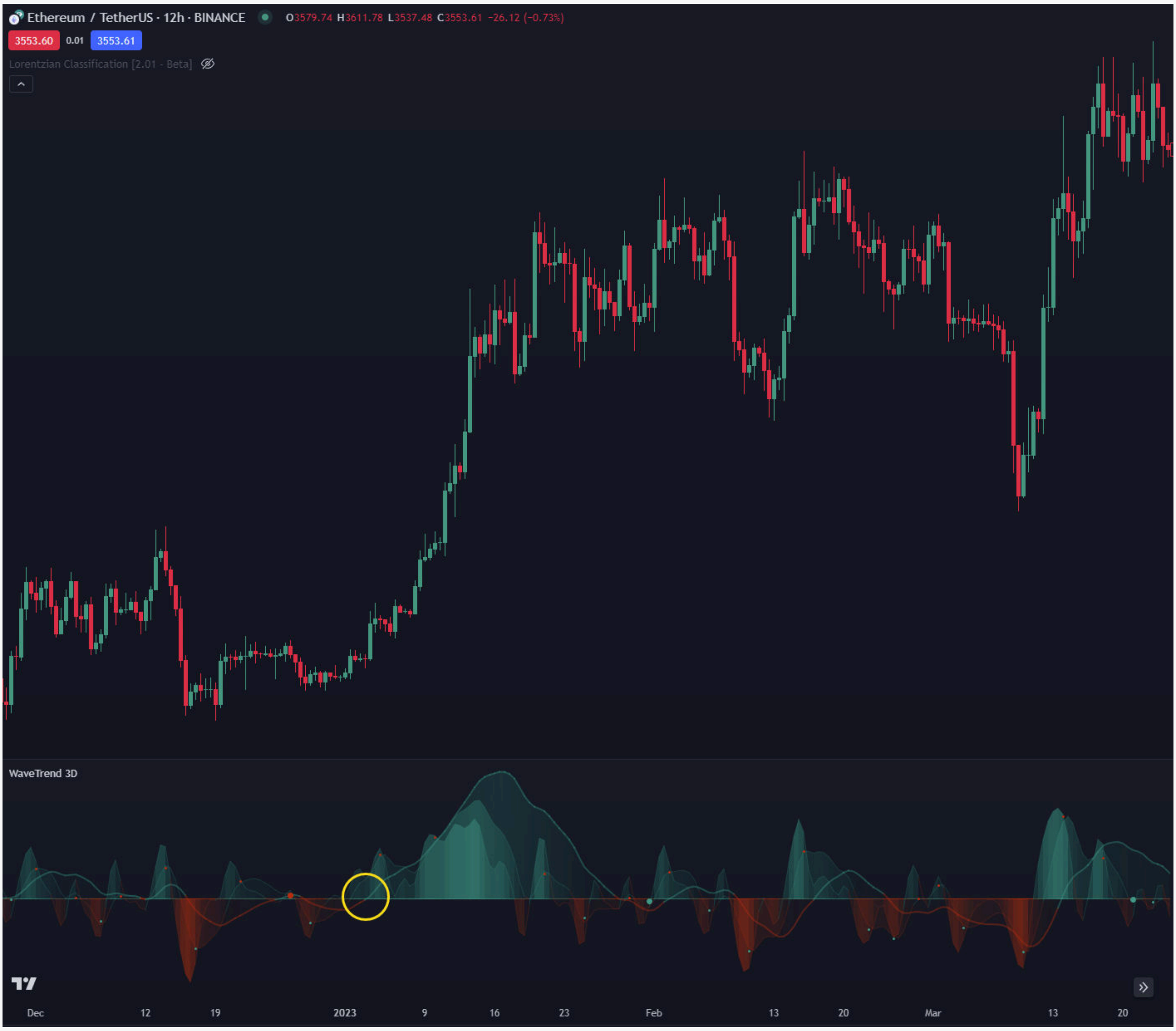
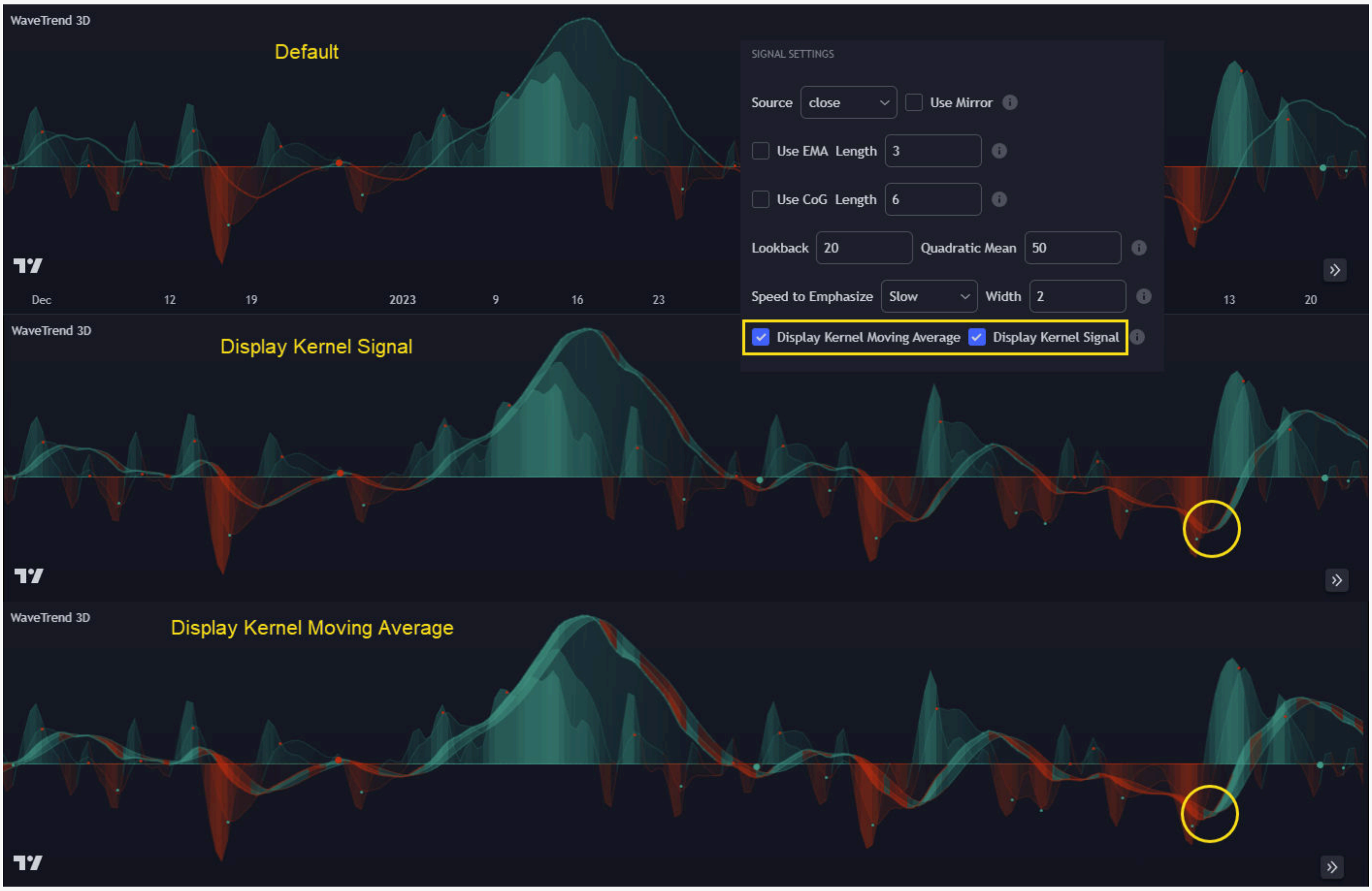

Ethereum / TetherUS · 12h · BINANCE
O3579.74 H3611.78 L3537.48 C3553.61 −26.12 (−0.73%)
3553.60
0.01
3553.61
Lorentzian Classification [2.01 - Beta]
WaveTrend 3D
Dec
12
19
2023
9
16
23
Feb
13
20
Mar
13
20

Fig 11: Slow oscillator zero-cross events typically mark significant shifts.

Kernel Estimators

Kernel estimators—particularly **Display Kernel Moving Average** or **Display Kernel Signal**—help detect trend bias with minimal lag.



SIGNAL SETTINGS

Source close □ Use Mirror

□ Use EMA Length 3

□ Use CoG Length 6

Lookback 20 Quadratic Mean 50

Speed to Emphasize Slow Width 2

☑ Display Kernel Moving Average ☑ Display Kernel Signal

Fig 12: A kernel ribbon changing colors can signal volatility spikes or market indecision.

Frequent color changes often correlate with a ranging market.

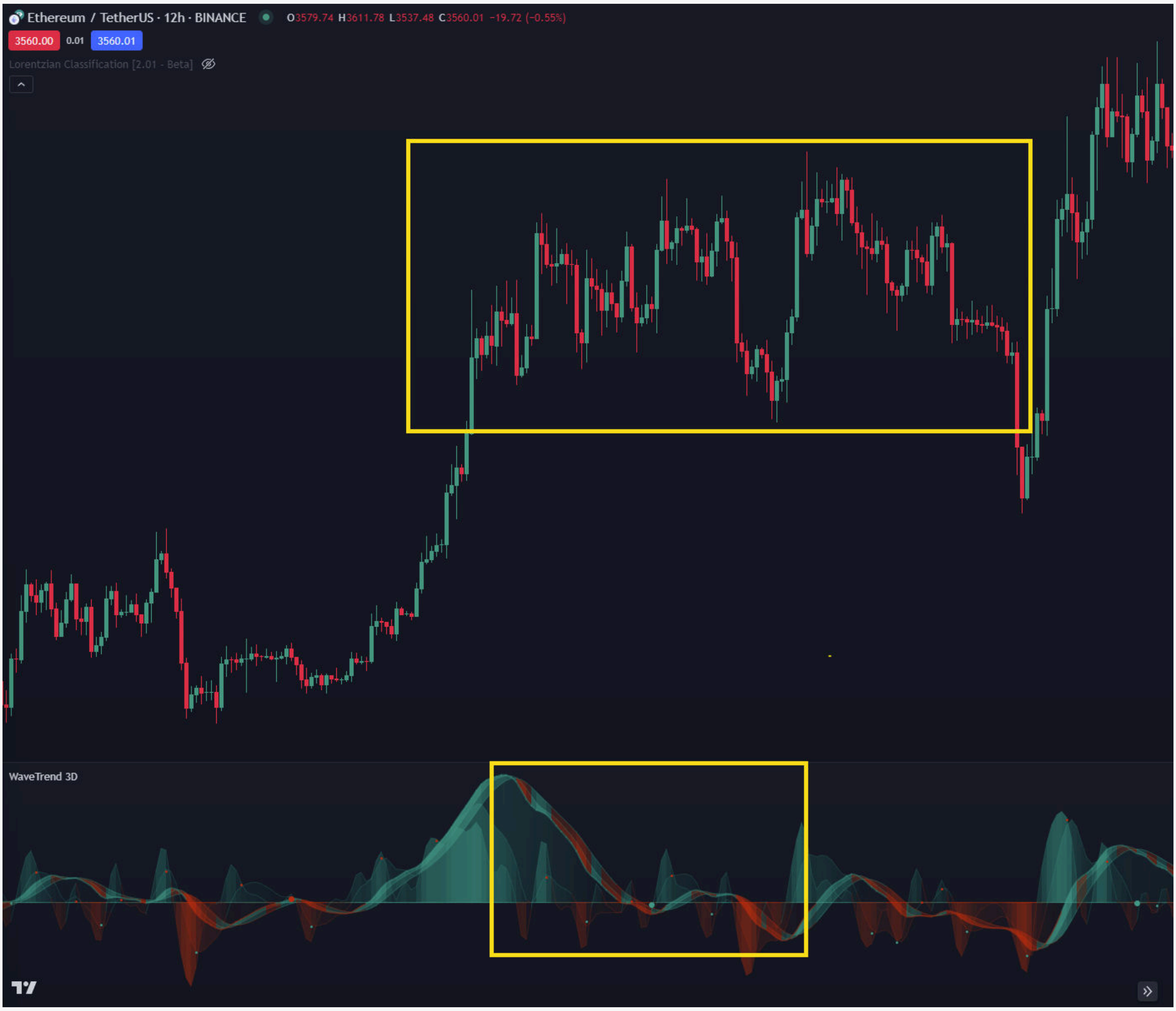
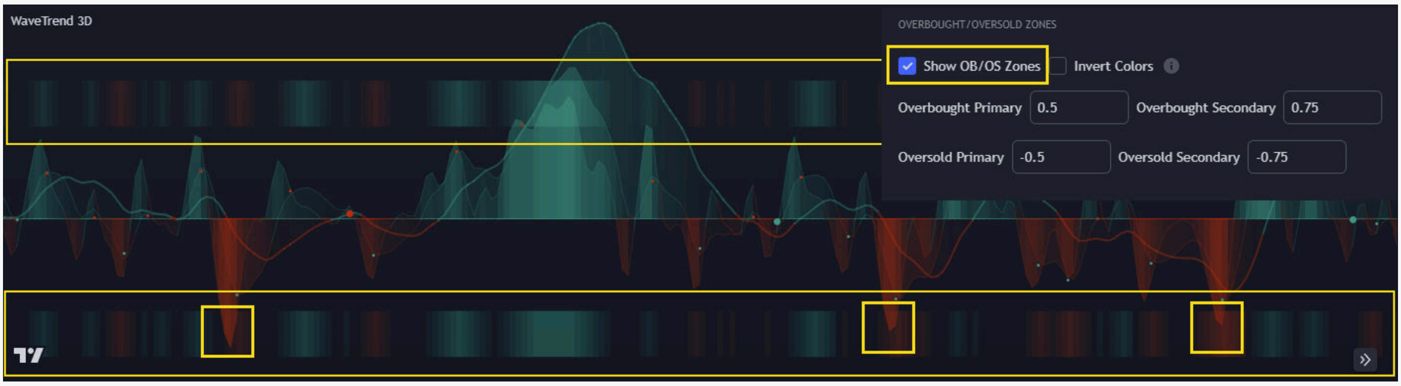


Fig 13: Rapid color shifts generally indicate sideways, choppy price action.

Overbought & Oversold Zones

Since Wavetrend 3D’s oscillators map between -1 and +1, zones around ±0.5 highlight extremes occurring in < 25% of conditions. These aren’t guaranteed “sell” or “buy” triggers—especially if the market transitions to a strong trend—but they’re valuable reversal zones.



OVERBOUGHT/OVERSOLD ZONES

☑ Show OB/OS Zones □ Invert Colors

Overbought Primary 0.5 Overbought Secondary 0.75

Oversold Primary -0.5 Oversold Secondary -0.75

Fig 14: Overbought/oversold thresholds mapped at ±0.5 within a -1 to +1 scale.

Practical Applications

1. **Trend Reversals**: Watch the **slow** oscillator crossing zero.
2. **Multi-Timeframe Confirmation**: Align small, normal, and slow oscillators for stronger signals.
3. **Divergences**: Large circle signals often front-run powerful reversals.
4. **Mirrored Mode Insights**: Identify smaller bullish or bearish “bubbles” inside a larger opposing wave.
5. **Risk Management**: Combine Wavetrend 3D with other analysis forms; trailing stop-losses can be keyed to major oscillator pivot points.

Frequently Asked Questions

Has Wavetrend 3D been ported to Python?

Yes. A Python implementation by Artnaz replicates the core logic of Wavetrend 3D. While it retains the multi-dimensional concept, remember that any backtesting or live trading environment may behave differently than Pine Script on TradingView.

Why are crosses plotted one candle prior to the actual cross?

It's mostly an aesthetic choice, keeping the circles aligned visually with their respective oscillator cross. If you prefer signals on the current bar, adjust the **Offset** to zero, but be aware that real-time bar developments can shift the final cross outcome.

Does Wavetrend 3D repaint?

Not in the conventional sense. Once a bar closes, the indicator locks its value. However, like any indicator, new incoming data on open bars can shift current oscillator readings. Rely on confirmed closes if repainting is a concern.

How do length and scaling settings affect each oscillator?

- **Scaling**: Adjusts how widely the oscillators fluctuate. Larger scaling increases the amplitude, making them more "sensitive"; smaller scaling narrows the range.
- **Length**: Defines the number of bars used in calculations. Longer lengths yield smoother lines with fewer signals; shorter lengths are more reactive and produce more frequent signals.

Future Pivots Indicator

WHAT YOU'LL LEARN

This comprehensive guide will introduce you to the Future Pivots (FP) indicator—a timing-focused tool that marks potential points of significant market movement. You'll discover how FP detects "price-time warping" and how it complements other systems like Lorentzian Classification or WaveTrend 3D.

Introduction

The **Future Pivots (FP)** indicator is a one-of-a-kind tool that zeroes in on **when** significant price action is likely to occur rather than predicting its directional outcome. By highlighting market “warp” moments—periods of abnormal volatility—FP helps traders prepare for pivotal moves, while leaving the ultimate bullish or bearish call to other indicators or your own analysis.

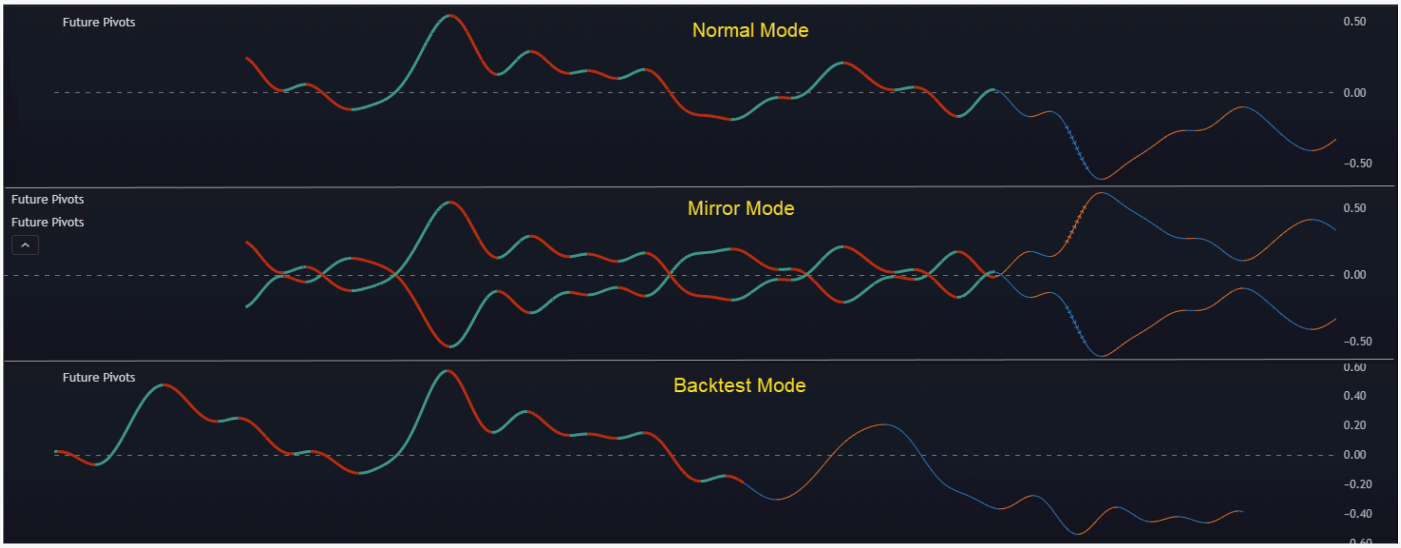


Fig 1: A high-level look at the FP indicator output on a chart.

Why This Matters

In fast-moving or volatile markets, traditional indicators often struggle to maintain accuracy when timing transitions. **Future Pivots** excels in these moments:

- **Better Preparedness:** Spot upcoming surges or collapses without committing to a bullish or bearish stance prematurely.

- **Enhanced Compatibility:** FP seamlessly integrates into almost any trading strategy, serving as a robust “timing confluence” tool.

How Future Pivots Works

Future Pivots harnesses advanced signal extraction and smoothing techniques (e.g., inverse Fisher transform, hyperbolic tangent normalization). These identify pivotal market frequencies—especially those aligning with the current timeframe’s “dominant cycle.”

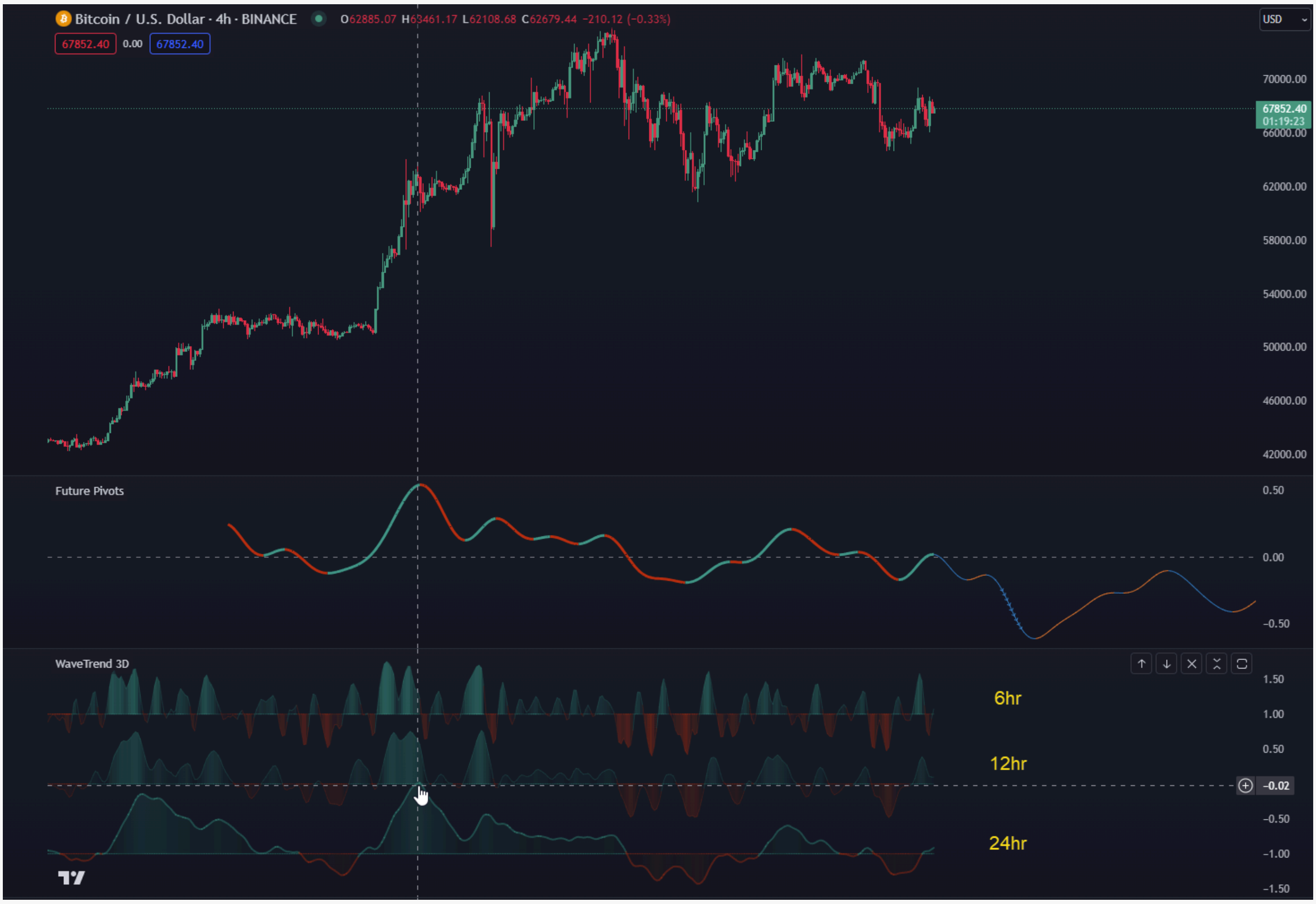


Fig 2: FP visualizing potential warp points (plotted in the future) on a 12-hour chart.

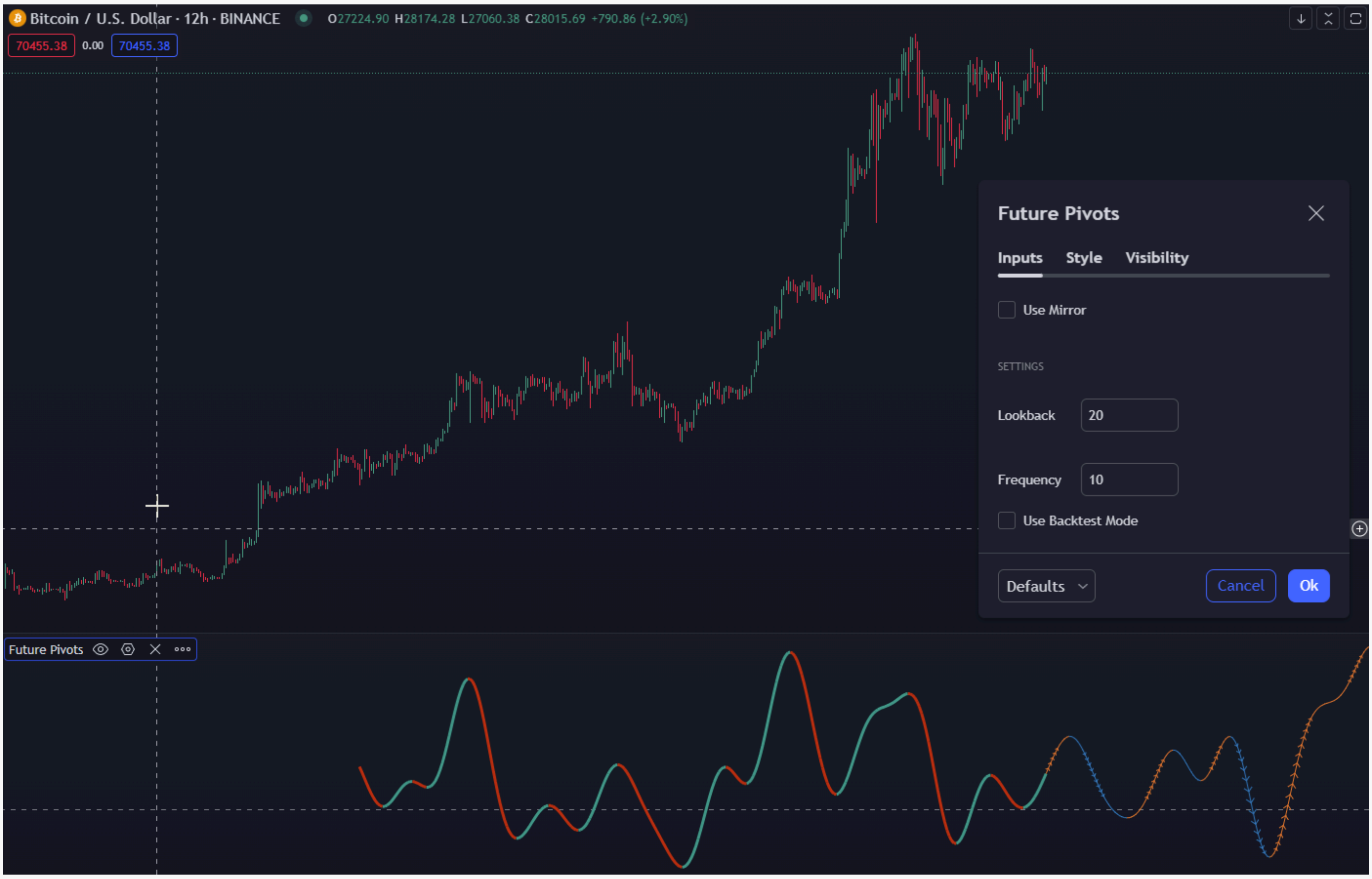
Sometimes, FP’s curve can appear like a truncated oscillator that projects ahead. This is **not** a directional prediction but a timing tool that flags spots for high potential volatility or momentum shifts.

What This Means for Traders

By focusing on “when” moves might happen rather than “which way,” traders can:

- **Time Trades More Precisely:** Use FP to know when a large move might occur, then consult indicators like Lorentzian Classification or WaveTrend 3D to gauge direction.

- **Adapt to “Warped” Markets:** FP detects moments of price-time distortion, helping traders avoid being blindsided by sudden volatility.
- **Utilize Flexible Settings & Backtesting:** Adjust lookback and frequency to refine pivot detection. The built-in backtest mode provides visual feedback for fine-tuning.


Bitcoin / U.S. Dollar · 12h · BINANCE
O27224.90 H28174.28 L27060.38 C28015.69 +790.86 (+2.90%)
70455.38
0.00
70455.38
Future Pivots
Inputs
Style
Visibility
Defaults
Cancel
Ok

Future Pivots

□ Use Mirror

SETTINGS

Lookback 20

Frequency 10

□ Use Backtest Mode

Fig 3: The main FP settings panel, offering lookback and frequency adjustments.

You can even spot chart patterns more clearly—like the inverse head and shoulders shown below—where the final color change often coincides with significant breaks to the upside or swift reversals.

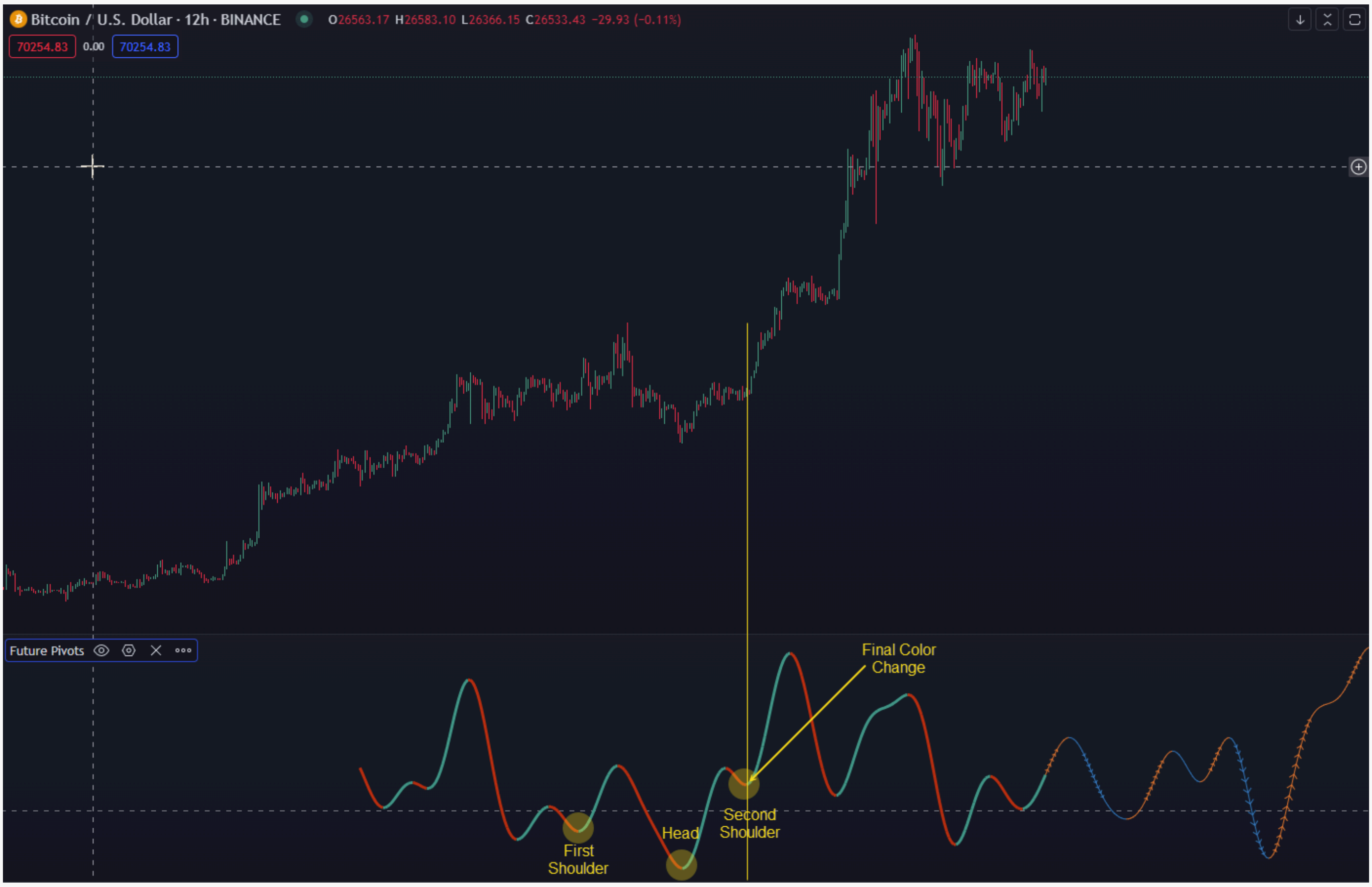


Fig 4: Example of an inverse head and shoulders pattern aligning with FP's final color change.

What This Guide Covers

In the sections that follow, you'll learn:

- **Core Concepts** like price-time warping, lookback frequency, pivot visualization, and mirrored views.
- **How to Set Up Backtest Mode** to evaluate FP's reliability on historical data.
- **Integration Tips** for combining FP with other indicators, including Lorentzian Classification.

KEY INSIGHT

While Future Pivots cannot directly feed into Lorentzian Classification due to technical differences, it still pairs beautifully as an additional layer of timing analysis.

Understanding Core Concepts

Future Pivots (FP) stands apart from many conventional oscillators by focusing on **when** markets might pivot rather than offering a price-direction forecast. Below, we explore the technical pillars that make FP such a valuable timing indicator.

Timing Over Direction

Why "Timing" Matters

Markets often blindside traders not merely by moving up or down, but by the *speed* and *unexpectedness* of that move. FP excels at uncovering these explosive pockets of volatility—allowing you to plan ahead and choose entry/exit strategies or complementary indicators to gauge the eventual direction.

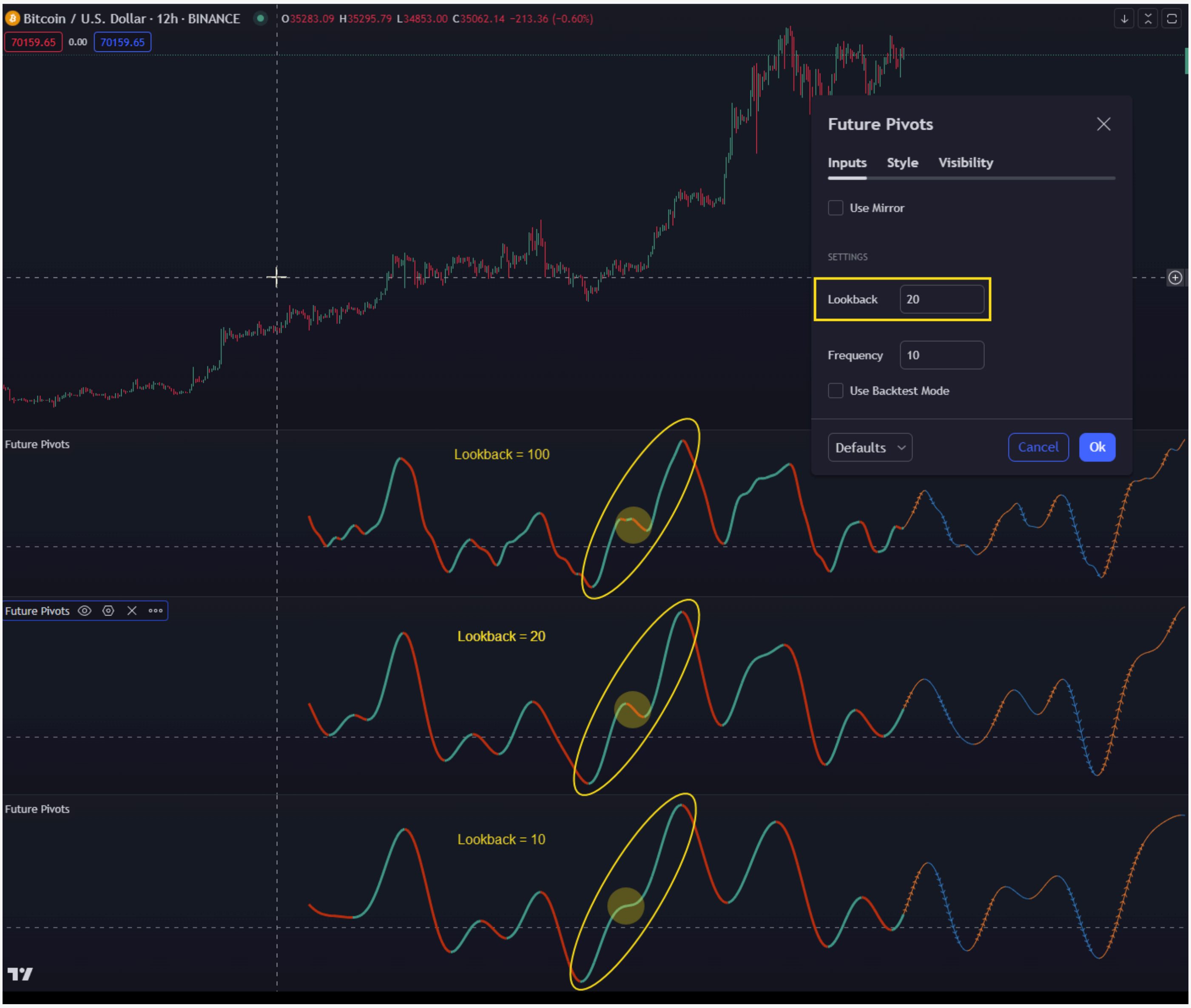
Compatibility with Other Systems

- **Lorentzian Classification**: While Lorentzian compares market points across warped space, FP flags the warping itself. Used together, they offer a potent combo of *where* (Lorentzian) and *when* (Future Pivots).
- **WaveTrend 3D**: FP can highlight pivot timing that often correlates with WaveTrend 3D's dominant frequency crossovers.

Price-Time Warping

Concept Clarification

Price-time warping describes periods of abnormally high/low volatility or momentum, during which standard linear indicators may falter. FP's internal algorithms detect these warping phases, prompting traders to be on high alert for potential breakouts or reversals.



Future Pivots

Inputs Style Visibility

□ Use Mirror

SETTINGS

Lookback 20

Frequency 10

□ Use Backtest Mode

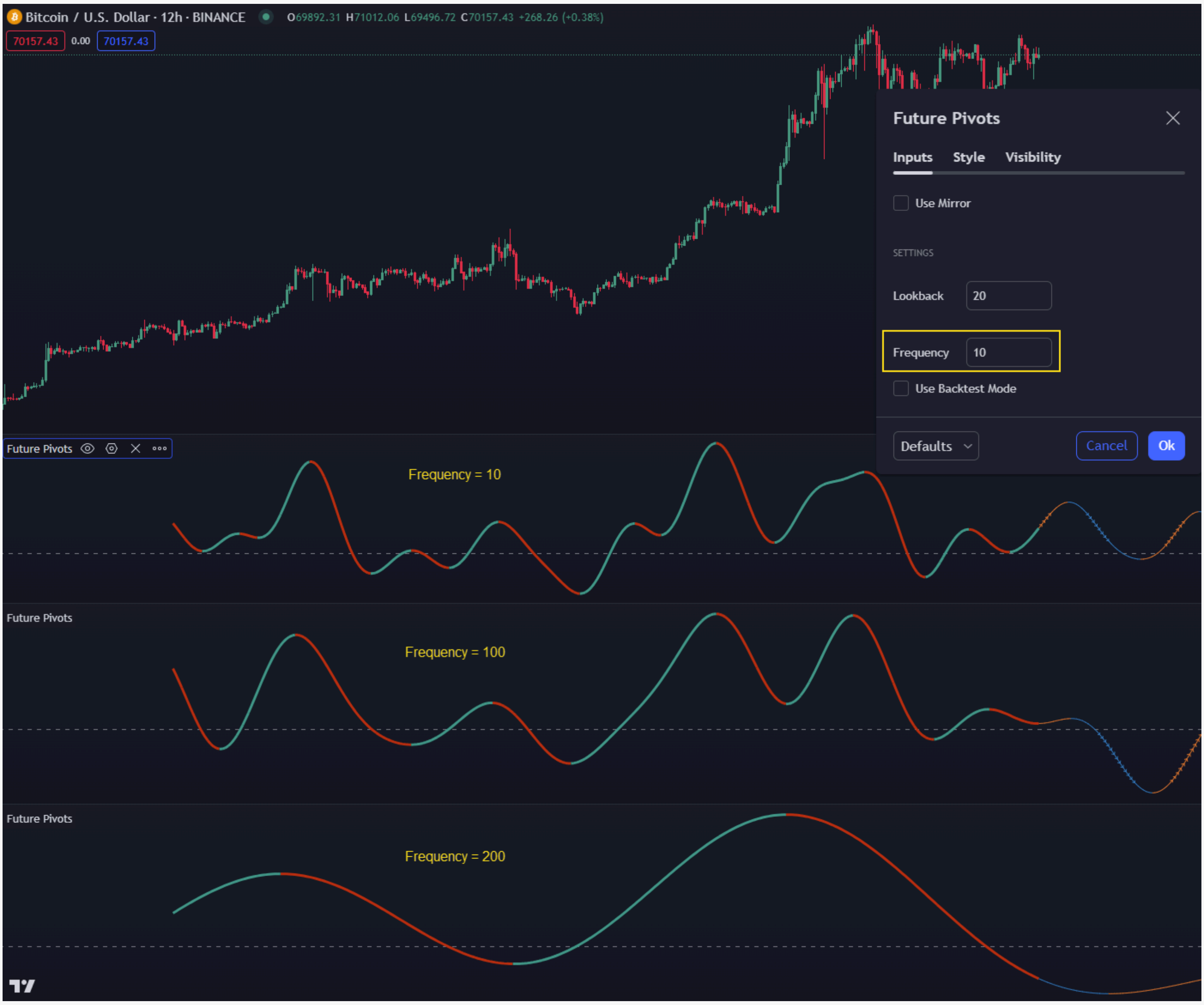
Defaults Cancel Ok

Fig 1: Lower lookback can reduce noisy oscillations in ranging markets, while higher lookback captures broader swings.

Lookback & Frequency Settings

Lookback

- **Granularity vs. Smoothing**: A lower lookback can filter out smaller fluctuations in sideways/ranging markets. A higher lookback (e.g., 100) may start to look like WaveTrend 3D, capturing broader oscillations more cleanly.



☐ Use Mirror

Lookback 20

Frequency 10

☐ Use Backtest Mode

Fig 2: Adjusting the frequency can shift pivot points, revealing different layers of market significance.

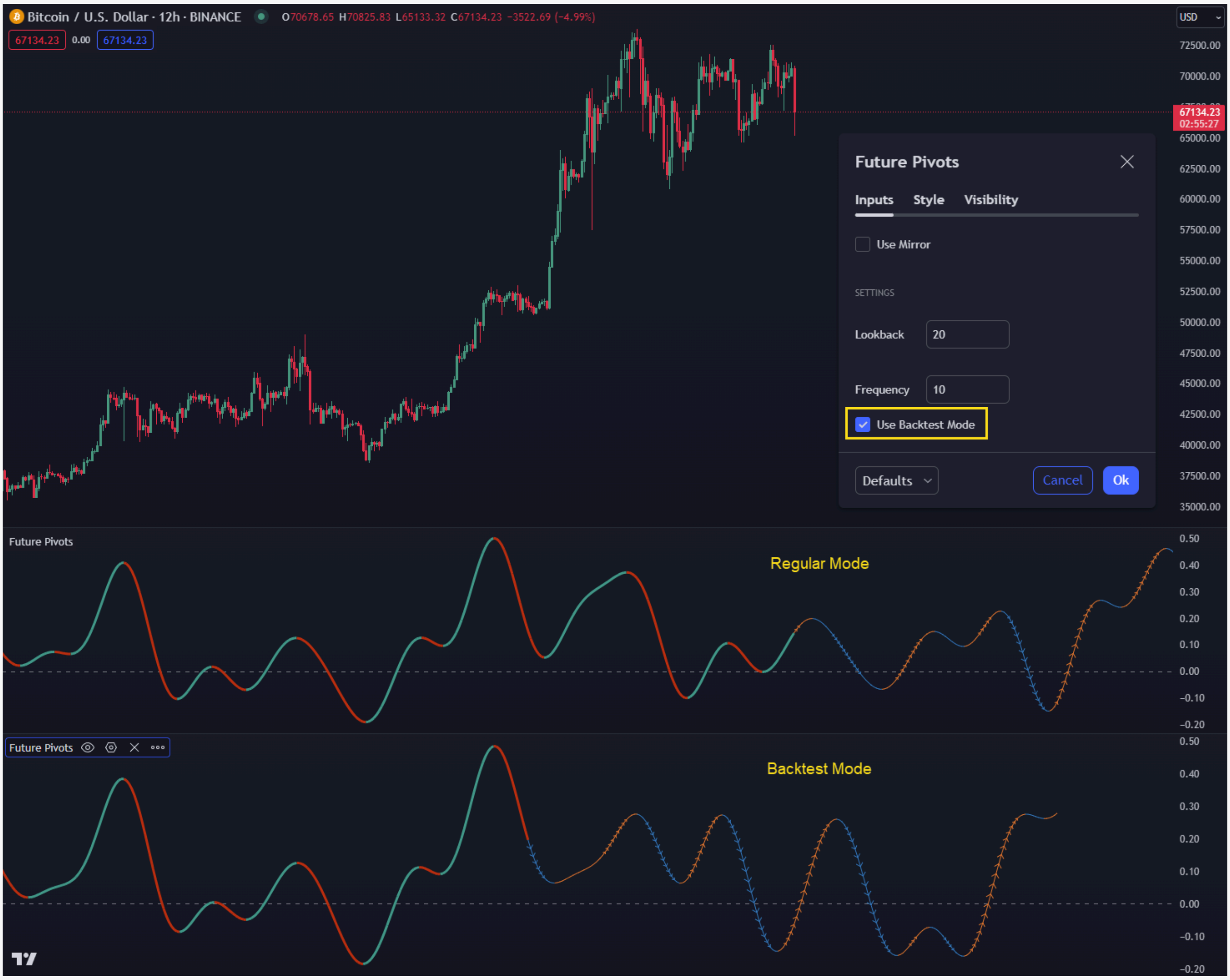
Frequency

- **Resembles Changing Timeframes**: Altering frequency might spotlight certain pivots that aren't as noticeable on default settings.

Backtest Mode

Visual Verification

The **Backtest Mode** preserves a chunk of historical bars (a "validation set") to show how your current FP settings might have performed in the past.



Future Pivots

Inputs Style Visibility

□ Use Mirror

SETTINGS

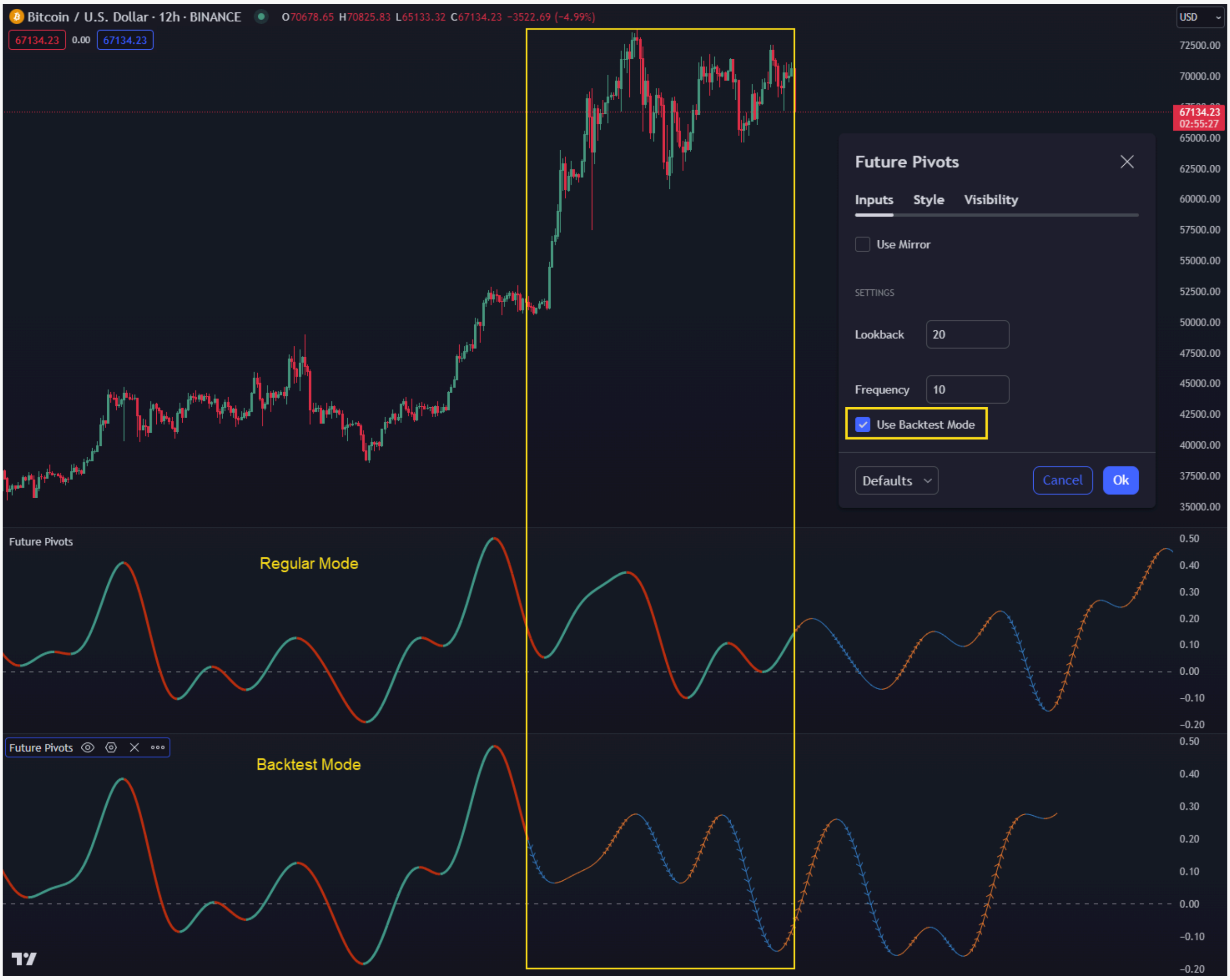
Lookback 20

Frequency 10

☑ Use Backtest Mode

Defaults Cancel Ok

Fig 3: Enable Backtest Mode to see how FP would have flagged past pivots.



Future Pivots

Inputs Style Visibility

□ Use Mirror

SETTINGS

Lookback 20

Frequency 10

☑ Use Backtest Mode

Defaults Cancel Ok

Fig 4: Reserved bars serve as a validation set, aiding a visual check of FP's reliability.

Look particularly for **color changes** (e.g., orange to blue) and **median crossings**—often correlating with future volatility spikes.

Mirrored Indicator View

Setting Up a Mirror

Mirroring FP provides a symmetrical, unbiased perspective. By duplicating the indicator and flipping one copy, you place pivot waves above and below a zero line:

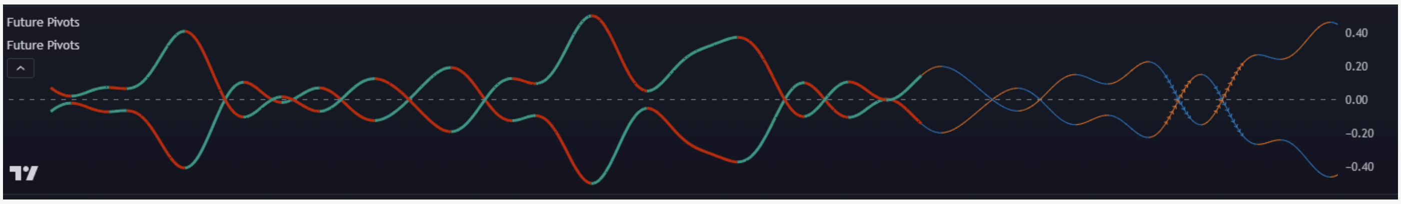


Fig 5: Mirroring FP can help you focus on pivot timing without directional bias.

1. **Duplicate & Drag**: Copy FP, then stack it on the same chart pane.

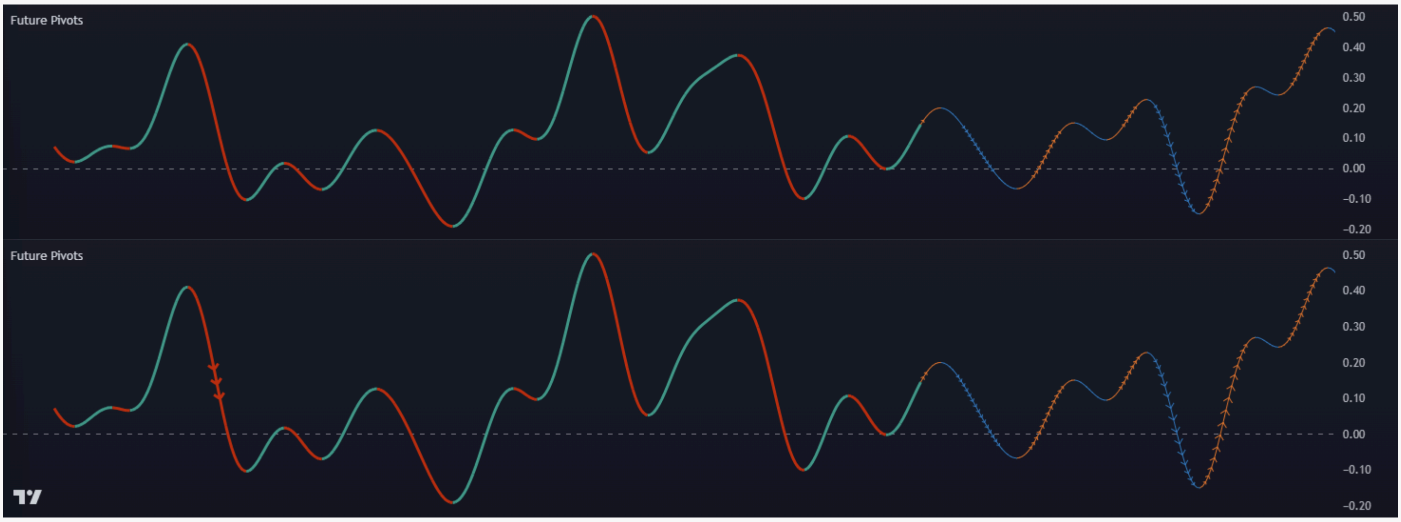
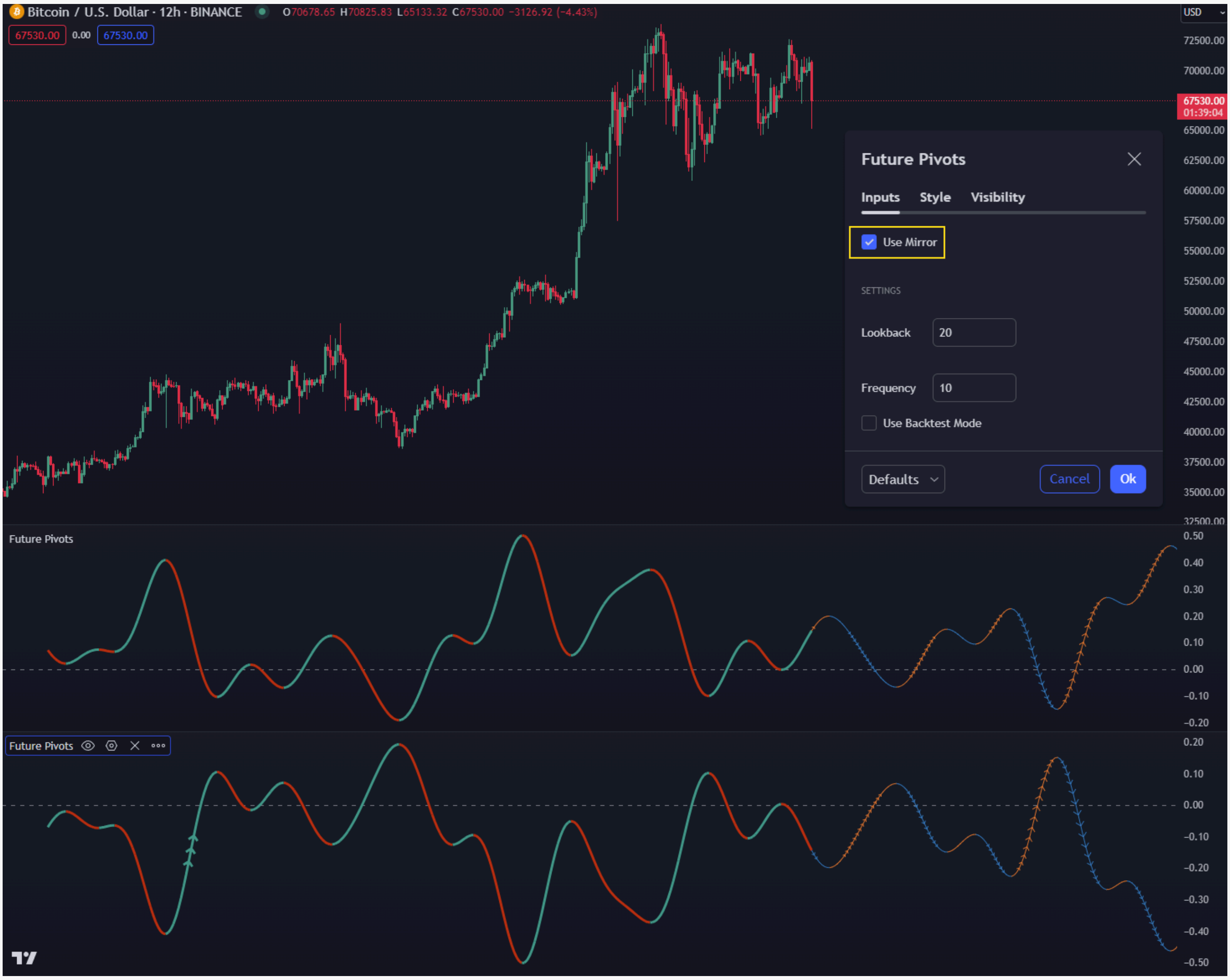


Fig 6: Duplicate Future Pivots and place it atop the original.

2. **Enable "Use Mirror"** in one of the copies.



Future Pivots

Inputs Style Visibility

☑ Use Mirror

SETTINGS

Lookback 20

Frequency 10

☐ Use Backtest Mode

Defaults Cancel Ok

Fig 7: The mirror toggle flips pivot signals across the median.

3. **Merge Panes**: Move both FP indicators into the same pane.

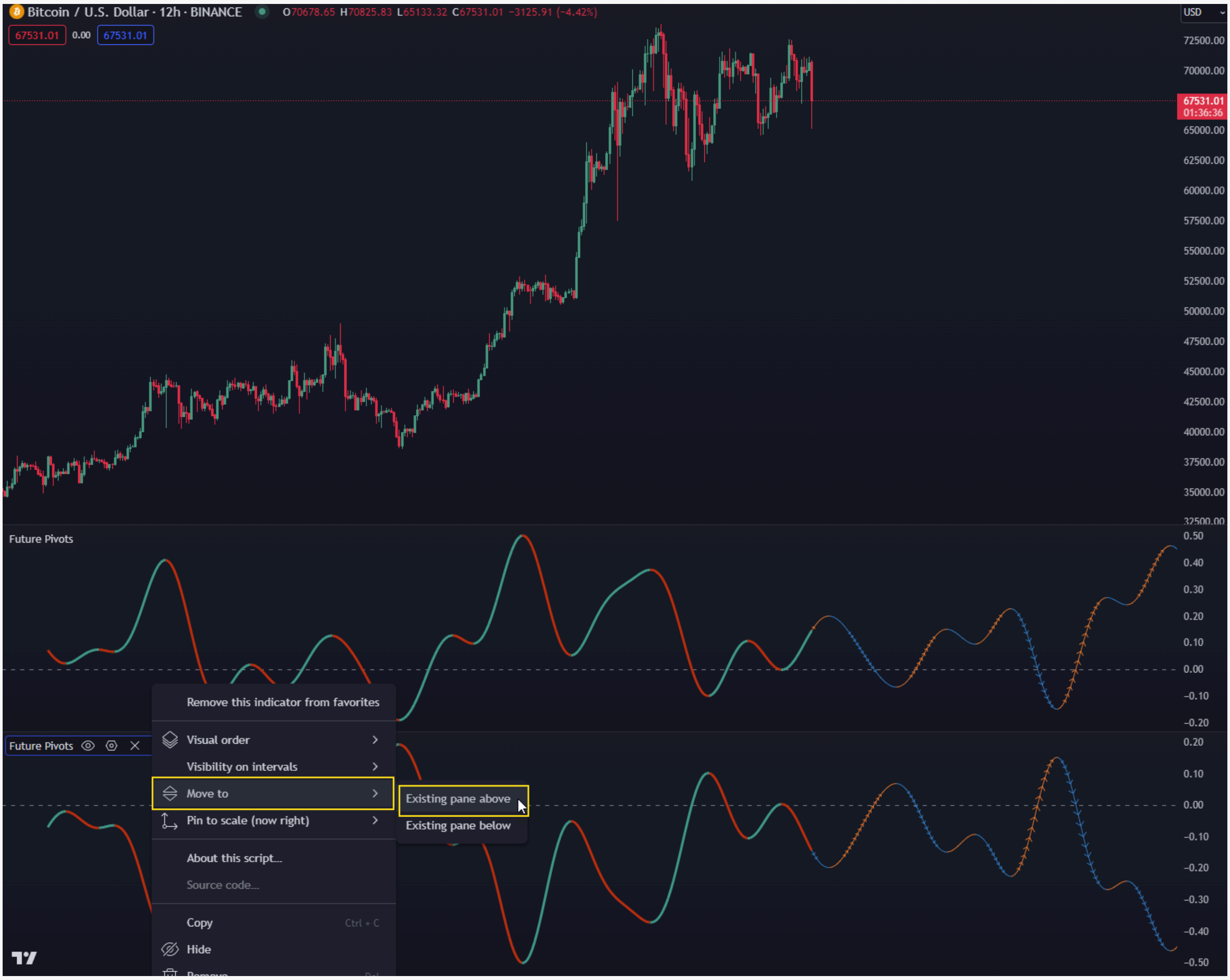


Fig 8: Combining multiple FP copies into a single pane for mirroring.

4. **Merge the Scales** on the right-hand side.

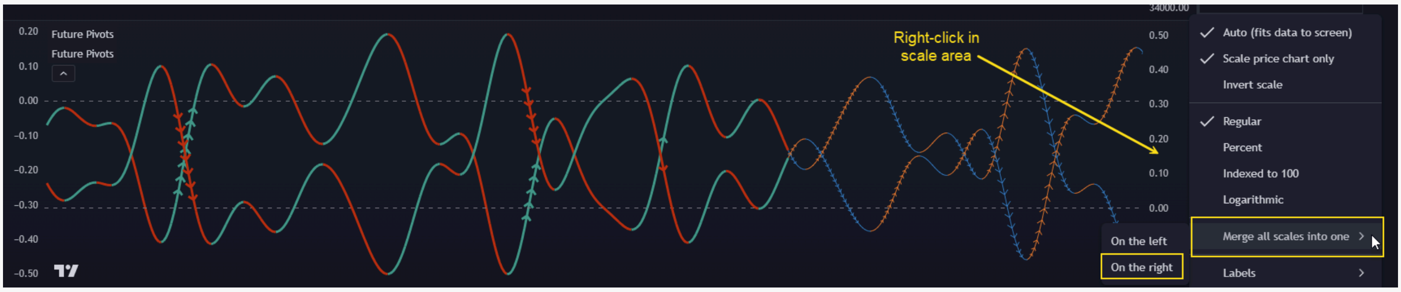


Fig 9: Ensuring both copies share a unified scale for consistent mirroring.

If done correctly, you'll see a symmetrical "above/below zero" version of FP:

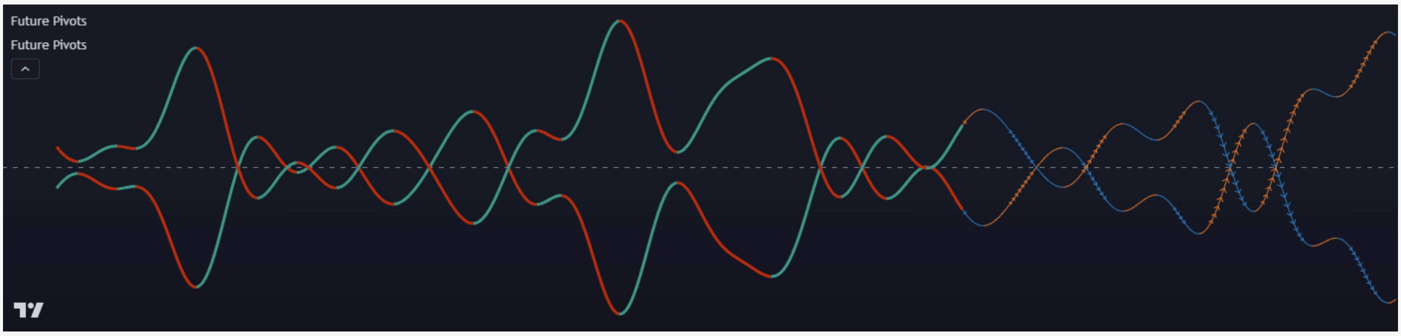


Fig 10: A fully mirrored FP setup, making pivot timing more visually intuitive.

Integration with Other Indicators

- **Lorentzian Classification**: Direct data feed isn't possible, but overlaying both on a chart can yield powerful timing+direction synergy.
- **WaveTrend 3D**: Combine FP's pivot windows with oscillator divergences or crossovers from WaveTrend 3D for robust trade signals.

Practical Applications

1. **Volatility Alerts**: Use median crosses or color changes to anticipate periods of high volatility or potential breakout.
2. **Trend Continuation vs. Reversal**: Let other direction-focused indicators confirm bullish/bearish bias after FP signals "it's time to move."
3. **Adaptive Strategy**: Adjust lookback or frequency as the market shifts between ranging and trending states.

Conclusion

Future Pivots brings a unique dimension to technical analysis by centering on *when* to pay closer attention to the market's next surge, reversal, or sideways phase. Through advanced smoothing, warping detection, and flexible settings, FP complements virtually any trading system. Understanding these **Core Concepts**—from price-time warping to mirrored views—will help you fully harness the indicator's capabilities and elevate your timing game.