

MCURVE OGX / GENIUS ONE Strategy & Indicator Master Runbook

MCURVE OGX / GENIUS ONE Strategy & Indicator Master Runbook

COLOR KEY: GREEN = Production / Long / Recommended RED = Archive / Short / Avoid / Risk ORANGE = Conditional / Warning CYAN = Research / Information PURPLE = Indicators / Filters

MCURVE OGX / GENIUS ONE

STRATEGY & INDICATOR MASTER RUNBOOK

Ranking • Logic • Risk • Alerts • Bot Deployment

2026-08-10 REVISION: Added V20.4.6 Execution Mode Truth Sync as the current #1 production candidate based on the latest observed 72.39% WR / 2.327 PF / \$927 max-DD / \$36,460 net snapshot, while marking forward alert/bot validation as still required.

RUNBOOK STATUS

Built from your strategy history, screenshots, pasted Pine logic, and the uploaded Performance Recovery source. Ranking emphasizes repeatability, profit factor, drawdown, trade count, alert reliability, and live-execution realism — not raw net profit alone.

1. Executive Ranking

HOW TO READ THE RANKING

The metrics below come from the screenshots/history you supplied and are not all from an identical test harness. Where a version was not independently isolated, the runbook marks that limitation instead of pretending the numbers are directly comparable.

CURRENT PRODUCTION SHORTLIST

V20.4.6 EXECUTION MODE TRUTH SYNC = current #1 production candidate, combining the V20.3 low-drawdown statistical profile with clearer bot/alert execution routing. V20.3 remains the untouched risk-adjusted control, V16 CLEAN remains the high-output control, and V15.0 remains the mature benchmark/core. Preserve all four as reference versions before building any new production branch.

#	Strategy	Score	WR	PF	Max DD
1	V20.4.6 Execution	9.9/10*	72.39%	2.327	\$927

#	Strategy	Score	WR	PF	Max DD
	Mode Truth Sync				
2	V20.3 MAE/MFE Immediate Failure Filter	9.8/10	72.28%	2.33	\$927
3	V16 CLEAN	9.7/10	72.47%	2.28	\$3,637.50
4	V15.0 Performan ce Recovery	9.5/10	72.36%	2.237	\$4,005
5	V15.1 Performan ce Recovery	9.3/10	71.13%	2.15	\$893.50
6	V15.5 PF Quality Edition	9.1/10	~72%	~2.3	~\$927*
7	V17 Adaptive Structure AI	8.6/10	Varies	Varies	Varies
8	V16.3 Run- Up Audit	8.4/10	Varies	Varies	Audit
9	EMA + MACD Momentum V4.3	8.0/10	Varies	~1.4 noted	Moderate
10	Trend Continuatio n / CHoCH Retest	7.8/10	Varies	~1.4 on 3m	Moderate
11	1-Min EMA Trend	7.5/10	Varies	Varies	Varies
12	MACD Confluence Dashboard	7.2/10	Varies	Varies	Varies
13	V19.1B OGX BUY-\$ELL\$	6.3/10	31.02%	1.154	\$24,480
14	Early Heikin- Ashi	6.0/10	84-95% HA	Unverified	Unverified

#	Strategy	Score	WR	PF	Max DD
	GENIUS ONE / OGX				
15	Ghost Feed / NASDAQ Elite Ghost Feed	5.8/10	Varies	Varies	Varies
16	Final Straw Confluence	5.5/10	Varies	Varies	Varies
17	GENIUS ONE V19.2 Diagnostic Trade Engine	4.0/10	Diagnostic	Diagnostic	Diagnostic

2. Strategy Runbook — Detailed Logic & Deployment

V20.4.6 — Execution Mode Truth Sync

Rank	Class	Score	Status
#1 Overall / Production Candidate	GREEN	9.9/10 Backtest & Architecture*	FORWARD-TEST / BOT VALIDATION

Observed performance: Latest MNQ 3-minute snapshot: 72.39% win rate, PF 2.327, max drawdown \$927, net profit \$36,460, 1,827 profitable trades out of 2,524.

Core logic

- Preserves the underlying V20.3/V15 institutional entry and exit engine while adding a cleaner execution-source distinction for automation.
- **EARLY + INTRABAR mode** is the faster execution route. It is intended to be paired with TradingView **alert() function calls only** when the goal is to let qualified early/intrabar signals reach the bot before the strategy fill route.
- **FILLS ONLY mode** is the strategy-sync route. It is intended to be paired with TradingView **Order fills only** so bot execution follows actual strategy order fills as closely as possible.
- The execution-mode work is designed to make BOT BUY/BOT SELL labeling more truthful by tying bot-facing labels to the selected execution source instead of treating every visual setup label as an executed bot event.

- The strategy-stat profile remained essentially unchanged from V20.3 after the routing changes, which is important because the objective was execution clarity rather than retuning the entry edge.

Why it ranks here

- Strength — Matches the strongest low-drawdown V20.3 profile while slightly improving the observed win count/net snapshot.
- Strength — PF remains above 2.3 with more than 2,500 trades in the observed test.
- Strength — \$927 max drawdown remains exceptional relative to the mature high-output benchmarks.
- Strength — Better separation between early/intrabar execution and strategy-fill execution makes live bot troubleshooting easier.
- Validation requirement — The 9.9/10 score is a backtest/architecture rating. Live-production confidence stays below 10/10 until forward testing confirms BOT label — TradingView Alert Log — webhook — bot receipt — expected order behavior without unexplained misses.
- Risk — Early/intrabar behavior can differ from historical bar behavior; forward testing is mandatory before funded automation.

Bot alert procedure

EXECUTION ROUTE

Use one route at a time. For the faster bot path, select **EARLY + INTRABAR** in the script and create the TradingView alert using **alert() function calls only**. For the strategy-matched path, select **FILLS ONLY** and create the TradingView alert using **Order fills only** with Message {{strategy.order.alert_message}}. Do not treat ordinary EARLY/BUY/SELL analysis labels as proof of bot execution; confirm the corresponding BOT-facing event in the Alert Log during forward testing.

V20.3 — MAE/MFE Immediate Failure Filter

Rank	Class	Score	Status
#2 Overall / Risk-Adjusted Control	GREEN	9.8/10	KEEP / TEST

Observed performance: Snapshot: 72.28% win rate, PF 2.33, max drawdown \$927, net profit \$36,310.50, 1,810 profitable trades out of 2,504.

Core logic

- Built on the V15/V16 institutional core but adds immediate-failure handling based on excursion behavior (MAE/MFE concept): identify trades that fail shortly after entry instead of allowing every setup to consume the full structural stop.

- Retains trend + VWAP + impulse + volume + candle-quality logic, then filters by market regime, adaptive flow, whipsaw protection, continuation quality, compression, and cooldown behavior.
- Main advantage is not a higher headline win rate; it is a major reduction in adverse excursion and drawdown while maintaining PF above 2.3.
- The best use case is live/bot deployment where preserving account buffer matters more than squeezing maximum historical net profit.

Why it ranks here

- Strength — Exceptional drawdown profile in the observed test.
- Strength — High trade count makes the sample more convincing than a low-trade optimized system.
- Strength — PF is the best observed among the mature high-trade versions.
- Strength — Architecture directly addresses the recurring problem of trades that go wrong immediately.
- Risk — Need the full source isolated and archived before changing anything.
- Risk — MAE/MFE logic must be verified in realtime because intrabar behavior can differ from historical bars.
- Risk — Do not compare its dollar net directly against V16 without identical contract size, date range, commissions, and slippage.

Bot alert procedure

ALERT ROUTE

Use STRATEGY ORDER FILLS as the production bot route. In TradingView Create Alert, select this strategy, choose Order fills (or Order fills only if available), set Message to `{{strategy.order.alert_message}}`, and send it to the bot webhook. Do not simultaneously send EARLY/intrabar alert() calls to the same execution endpoint unless the bot de-duplicates them.

V16 CLEAN

Rank	Class	Score	Status
#2 Overall / #1 High-Output	GREEN	9.7/10	KEEP / TEST

Observed performance: Snapshot: 72.47% win rate, PF 2.28, max drawdown \$3,637.50, net profit \$144,330, 1,548 profitable trades out of 2,136.

Core logic

- Cleaned production branch of the V15 architecture: fast/slow EMA trend, session VWAP, impulse breakout, volume confirmation, candle body quality, and regime gating.

- Focuses on removing redundant/dead logic so the signal path is easier to audit and live execution is more deterministic.
- TP1/TP2 + break-even + runner structure remains the main payoff engine, allowing the strategy to win frequently while still capturing continuation.
- Compared with V20.3, V16 appears to prioritize throughput and profit capture over minimum excursion.

Why it ranks here

- Strength — Highest observed net profit among the strongest clean snapshots.
- Strength — Strong WR and PF despite >2,000 closed trades.
- Strength — Cleaner signal path is preferable for bot troubleshooting.
- Strength — Excellent benchmark for future experiments.
- Risk — Drawdown is higher than V20.3/V15.1.
- Risk — Need identical commission/slippage testing before declaring it superior live.
- Risk — Do not re-add every historical filter; CLEAN's simplicity is part of its advantage.

Bot alert procedure

ALERT ROUTE

Recommended bot mode: strategy order fills only. Use `{{strategy.order.alert_message}}` in the alert Message field. If the source also contains `alert()` early signals, create a separate notification-only alert for those rather than sending them to the execution webhook.

V15.0 — Performance Recovery Edition

Rank	Class	Score	Status
#3 Overall / Benchmark Core	GREEN	9.5/10	KEEP / TEST

Observed performance: Snapshot: 72.36% win rate, PF 2.237, max drawdown \$4,005, net profit \$138,485, 1,524 profitable trades out of 2,106.

Core logic

- Core defaults: 2 contracts; 20-point SL; TP1 10; TP2 25; break-even trigger 6; trail trigger 15; trail distance 7; EMA 5/13; ATR 10; volume MA 10 with 1.15 spike multiplier.
- Engine 1 scores trend (40), VWAP location (20), and impulse (40). Engine 2 independently scores EMA direction (35), volume > volume MA (25), and bullish/bearish candle body (40).
- Session logic labels 09:30–11:00 ET as HOT, 13:30–15:00 as GOOD, 12:00–13:30 as DEAD, and other periods NORMAL.

- Market compression uses EMA spread, ATR contraction, VWAP overlap, low volume, inside bars, and tight range. Trend persistence counts price vs EMA/VWAP plus directional closes and EMA slope.
- A+ scoring layers EMA spread, volume, impulse, VWAP expansion, candle quality, trend persistence, session score, and optional HTF alignment; compression subtracts quality.
- Market Regime classifies UNSTABLE / CHOP / TRANSITION / TRENDING. Candidate entries then pass regime, adaptive flow, whipsaw guard, adaptive cooldown, trade refinement, late-expansion, continuation quality, compression, and weak-session gates.
- Exit engine uses TP1/TP2 split, break-even after +6 points, runner trailing logic, and optional smart runner exit/profit lock (off by default in this edition).

Why it ranks here

- Strength — Most mature and explainable full architecture.
- Strength — Near-V16 performance confirms the core is not a one-off result.
- Strength — Excellent base for research because every filter has a clear role.
- Strength — Bot routing includes early, intrabar, and fill priorities.
- Risk — Complexity can create hidden dead code or redundant gating.
- Risk — calc_on_every_tick plus intrabar alert paths require careful live-vs-backtest validation.
- Risk — Using ALL alert modes to the same execution endpoint can create duplicate order risk.

Bot alert procedure

ALERT ROUTE

For safest automation, set script Alert Mode to FILLS ONLY (or disable early/intrabar execution) and create a TradingView alert on Order fills with Message `{{strategy.order.alert_message}}`. If you intentionally want early entries, use Any alert() function call as a separate execution mode — never both without bot-side setup IDs/de-duplication.

V15.1 — Performance Recovery

Rank	Class	Score	Status
#4 Overall / Low-DD Benchmark	GREEN	9.3/10	KEEP / TEST

Observed performance: Snapshot: 71.13% win rate, PF 2.15, max drawdown \$893.50, net profit \$26,809, 1,518 profitable trades out of 2,134.

Core logic

- Same Performance Recovery family with runner-friendly behavior and a softer filter stack than over-restricted prior branches.
- Preserves adaptive flow and market regime selection while avoiding overly aggressive smart runner exits by default.
- Primary value is the unusually low observed drawdown while still preserving a large trade sample and PF >2.

Why it ranks here

- Strength — Very strong risk efficiency.
- Strength — More than 2,000 trades in snapshot.
- Strength — Excellent candidate for funded-account conservative profile.
- Risk — Lower net than V16/V15.0; may leave money on the table.
- Risk — Need isolated contract sizing and date-range comparison to V20.3.

Bot alert procedure

ALERT ROUTE

Same production rule as V15.0: prefer strategy fills and {{strategy.order.alert_message}}.
Early BUY/SELL should be notification-only unless explicitly testing early execution.

V15.5 — PF Quality Edition

Rank	Class	Score	Status
#5 Research Candidate	BLUE	9.1/10	KEEP / TEST

Observed performance: Observed dashboard showed ~72% WR / ~2.3 PF / ~\$927 DD / ~\$36K net, but that snapshot matched the V20.3 run; treat as unverified until isolated.

Core logic

- Adds a secondary impulse-exhaustion reversal engine to the V15 core.
- Big-move detection uses lookback and minimum point excursion. Reversal quality examines indecision body percentage, hammer/inverse-hammer wick ratio, EMA confirmation, optional VWAP/CHoCH confirmation, regime cleanliness, and flow agreement.
- Impulse quality score awards points for prior large move, hammer structure, EMA confirmation, clean regime, flow agreement, and breaking the prior bar.
- PF controls can require minimum market quality, stronger impulse score, block strong opposite flow, avoid weak sessions, and enable Higher PF Mode.

- Impulse add-on can operate as CONFIRMATION ONLY or SECONDARY ENTRY, making it either a filter or an additional trade generator.

Why it ranks here

- Strength — Good way to capture exhaustion reversals missed by pure continuation logic.
- Strength — Built-in audit counters help identify dead filters and whether the add-on genuinely changes entries.
- Strength — Higher-PF switches are explicit rather than hidden.
- Risk — Secondary entry logic increases overfit risk.
- Risk — Screenshot performance is not yet independently attributable to this version.
- Risk — Reversal logic after large impulses is more regime-sensitive than continuation logic.

Bot alert procedure

ALERT ROUTE

If SECONDARY ENTRY is enabled, keep the production alert route on strategy fills so both original and impulse entries are handled consistently. Use `{{strategy.order.alert_message}}`. IMP BUY/SELL visual labels do not need separate executable alerts unless intentionally testing the add-on alone.

V17 — Adaptive Structure AI

Rank	Class	Score	Status
#6 Research	BLUE	8.6/10	KEEP / TEST

Observed performance: Historical notes: adaptive AUTO behavior; pivot_len 3; lookback 40; EMA 21/48 structure; ADX; max SL 30; TP splits; multiple BE modes.

Core logic

- Attempts to adapt entry strictness and risk to structure instead of using a single fixed condition set.
- Uses pivots/BOS/CHoCH-style structural context, EMA trend structure, ADX strength, and adaptive stop/BE logic.
- Designed to shift between continuation and reversal contexts rather than assume one regime.

Why it ranks here

- Strength — Conceptually sophisticated regime adaptation.
- Strength — Structure-aware rather than indicator-only.
- Strength — Useful research framework for comparing dynamic vs fixed filters.

- Risk — High parameter count increases optimization risk.
- Risk — Adaptive logic can become difficult to reason about live.
- Risk — Later CLEAN/MAE-MFE branches produced stronger observed statistical evidence.

Bot alert procedure

ALERT ROUTE

Prefer strategy order fills. If the script exposes alert() early/standard signals, keep them separate from the execution webhook. Use fill alerts as the source of truth for bot position state.

V16.3 — Run-Up Audit

Rank	Class	Score	Status
#7 Research / Diagnostics	BLUE	8.4/10	KEEP / TEST

Observed performance: Known defaults: SL 20, TP1 10, TP2 25, BE 6, trail 15/7; EMA 5/13; ATR 10; volume MA 10 x 1.15; min score 75.

Core logic

- Built to audit run-up, drawdown, entry quality, and whether the strategy is giving back too much after favorable excursion.
- Uses the mature V15-style signal engine but emphasizes analytics around trade path rather than only final P/L.

Why it ranks here

- Strength — Excellent for finding where winners turn into losers.
- Strength — Useful for designing MAE/MFE failure filters and runner locks.
- Strength — Good diagnostic companion to V16/V20.
- Risk — Audit variants can distort production behavior if diagnostic logic is allowed to execute.
- Risk — Do not confuse analytics labels with actual fills.

Bot alert procedure

ALERT ROUTE

For research alerts, use fills only. Do not send diagnostic/audit labels to the bot. The bot should respond only to explicit executable strategy orders.

EMA + MACD Momentum V4.3

Rank	Class	Score	Status
#8 Benchmark	ORANGE	8.0/10	KEEP / TEST

Rank	Class	Score	Status
------	-------	-------	--------

Momentum Strategy

Observed performance: Historical note: PF around 1.4 on some 3-minute tests; ATR stop 1.60 with 30-point cap; BE 12/1; optional fixed TP 40.

Core logic

- Trend direction comes from EMA structure; entry requires MACD momentum alignment and price interaction with the 9 EMA.
- Continuation entries are allowed when momentum re-accelerates rather than only on the first cross.
- Exits use histogram fade/EMA breaks plus ATR-based risk and optional fixed target.
- Break-even activates only after meaningful favorable movement to avoid premature scratches.

Why it ranks here

- Strength — Simple and explainable.
- Strength — Good benchmark against more complex MCURVE engines.
- Strength — MACD histogram provides useful momentum timing.
- Risk — MACD lags in very fast NQ reversals.
- Risk — Can overtrade when EMA and MACD oscillate around zero in chop.
- Risk — Needs session/regime filter to compete with later versions.

Bot alert procedure

ALERT ROUTE

If coded as a strategy, use Order fills and `{{strategy.order.alert_message}}`. If coded with `alertcondition()` only, create separate LONG and SHORT alerts from the named conditions and send bot JSON such as action BUY/SELL, symbol, contracts, SL/TP.

Trend Continuation / CHoCH Retest

Rank	Class	Score	Status
------	-------	-------	--------

#9 Structure Specialist	ORANGE	7.8/10	KEEP / TEST
-------------------------	--------	--------	-------------

Observed performance: Historical note: PF around 1.4 on 3-minute; impulse + CHoCH retrace model; MTF 5/15/30 optional.

Core logic

- Wait for an impulsive directional move, then a CHoCH/retracement into structure.

- Entry occurs on indecision, hammer/inverse-hammer, or renewed directional confirmation after the retrace.
- Optional multi-timeframe filters align 5m/15m/30m structure and breakout direction.
- One-signal-per-trend logic prevents repeated entries in the same directional leg.

Why it ranks here

- Strength — Closer to actual market structure than pure oscillator entries.
- Strength — Good for strong directional NQ sessions.
- Strength — Naturally limits chasing if retrace requirement is enforced.
- Risk — CHoCH definitions are highly implementation-sensitive.
- Risk — Can miss V-shaped continuation without retrace.
- Risk — Low-timeframe pivot logic can repaint if not confirmed.

Bot alert procedure

ALERT ROUTE

Use bar-close confirmed strategy fills for production. If CHoCH is calculated with confirmed pivots, alert on order fills; if it is intrabar, keep alert() signals notification-only until forward-tested.

1-Min EMA Trend

Rank	Class	Score	Status
#10 Trend Following	ORANGE	7.5/10	KEEP / TEST

Observed performance: Historical note: 9 EMA trend; TP progression through 21/30/48 EMA; 10-minute tests looked cleaner; daily trades targeted under 20.

Core logic

- 9 EMA is the fast directional trigger; broader EMA stack defines trend context.
- Profit management references deeper EMA structure rather than only fixed points.
- Exit conditions include indecision and reversal candle structure such as red close below key EMA / inverse hammer in the opposite direction.
- BOS/CHoCH checks can be layered to avoid taking every EMA touch.

Why it ranks here

- Strength — Visual and easy to trade manually.
- Strength — Works well as a trend-state overlay even when not used for entries.
- Strength — Good educational benchmark.

- Risk — 1-minute NQ noise can create false flips.
- Risk — EMA-only logic is vulnerable to lunchtime chop.
- Risk — Performance was better on slower timeframe, implying the name can be misleading.

Bot alert procedure

ALERT ROUTE

For bot trading, use the slower confirmed version and strategy fills. Avoid intrabar 1-minute alert() execution unless the bot has strict cooldown and one-position locks.

MACD Confluence Strategy Dashboard

Rank	Class	Score	Status
#11 Confluence/Scoring Strategy	ORANGE	7.2/10	KEEP / TEST

Observed performance: Known weights: EMA 20, MACD 15, Histogram 10, Slope 10, VWAP 10, Pullback 10, Candle 10, Distance 5, Zero 5, Session 5. 2-minute chart tested better than other TFs.

Core logic

- Scores multiple independent features into a weighted long/short confluence percentage.
- EMA trend and MACD line/histogram are the primary directional components; VWAP, pullback, candle body, distance, zero-line context, and session quality refine entries.
- Typical risk inputs used TP 30, SL 20, BE trigger 15/lock 1, and entry confluence around 55.

Why it ranks here

- Strength — Transparent scoring makes debugging easy.
- Strength — Useful dashboard for manual confirmation.
- Strength — Can be tuned by weight rather than adding dozens of hard gates.
- Risk — Weighted scores can hide correlated inputs — EMA, MACD, histogram, slope all partly measure the same thing.
- Risk — 2-minute advantage may be sample-specific.
- Risk — Past versions had no-trade/report issues during development.

Bot alert procedure

ALERT ROUTE

If strategy orders exist, use order fills. If the dashboard is indicator-only, create alerts on finalBuy/finalSell conditions — not raw score changes — and include a cooldown/position state in the bot.

V19.1B — OGX BUY-\$ELL\$

Rank	Class	Score	Status
#12 / Retire as Primary, Keep for Components	RED	6.3/10	ARCHIVE / DIAGNOSTIC

Observed performance: Snapshot: 31.02% WR (309/996), PF 1.154, max drawdown \$24,480, net profit \$75,728; chart showed 16-contract configuration.

Core logic

- Adaptive channel uses a long ATR-derived width; buys around lower-mid recovery and sells around upper-mid rejection with strict candle body filtering.
- Pro Reversal Engine adds lowest/highest sweeps, engulfing candles, and MACD direction around the channel.
- OGX FLIP detects channel breaks and grades them with volume + impulse confirmation into WEAK or STRONG flips.
- Risk engine supports fixed, ATR, micro-scalp, multi-TP splits, Smart TP2 exhaustion, and many contract presets.
- Bot JSON builder is one of the strongest parts: it exports action, symbol, contracts, split preset, timeframe, absolute price/SL/TP, TP quantities, runner configuration, and Smart TP state.
- Same-bar execution lock and hard-truth controls were added to reduce duplicate/preview confusion.

Why it ranks here

- Strength — Excellent bot/export architecture.
- Strength — Useful OGX FLIP and opening-range components.
- Strength — Flexible split/risk presets.
- Risk — Observed risk-adjusted stats are weak: PF barely above 1 and DD very large.
- Risk — 31% WR requires strong payoff asymmetry that the PF does not sufficiently show.
- Risk — 16-contract test inflates dollar results and risk.
- Risk — Too many execution/risk modes complicate production validation.

Bot alert procedure

ALERT ROUTE

This script explicitly supports Bot Alert Mode = Strategy Order Fills or alert() Immediate. Production recommendation: select Strategy Order Fills, create TradingView alert on Order fills, set Message to {{strategy.order.alert_message}}, and use the bot webhook URL. Do not also enable alert() Immediate to the same bot.

Early Heikin-Ashi GENIUS ONE / OGX Variants

Rank	Class	Score	Status
#13 / Historical Research	ORANGE	6.0/10	KEEP / TEST

Observed performance: History: 84–95% WR reported on HA versions; raw-candle conversion dropped dramatically (~45% and lower with slippage in some tests).

Core logic

- Heikin-Ashi candle smoothing was used to identify cleaner trend, pullback, and momentum sequences.
- Signals often appeared earlier and looked visually excellent because HA transforms OHLC values and suppresses noise.
- Several variants combined EMA structure, impulse, candle patterns, and early BUY/SELL signals.

Why it ranks here

- Strength — Excellent visual trend recognition.
- Strength — Useful as a regime/confirmation overlay.
- Strength — Can reduce emotional noise for manual reading.
- Risk — HA prices are synthetic; strategy fills on HA can produce unrealistic backtest assumptions if execution is supposed to occur on raw market prices.
- Risk — Historical labels disappearing/reappearing after layout changes created confidence issues.
- Risk — Large raw-vs-HA performance gap makes it unsuitable as the production truth source.

Bot alert procedure

ALERT ROUTE

Do not execute a bot directly from synthetic HA fills unless the strategy explicitly calculates signals on HA but places orders using raw-symbol prices. Safer: use HA as confirmation and execute a raw-candle strategy-fill alert.

Ghost Feed / NASDAQ Elite Ghost Feed

Rank	Class	Score	Status
#14 Specialist	ORANGE	5.8/10	KEEP / TEST

Observed performance: Historical logic: volume spike around 2.2x, VWAP mean-reversion distance, reversal context.

Core logic

- Looks for abnormal participation/volume and price displacement away from VWAP or structure.
- Attempts to capture reversal/mean-reversion when the move appears exhausted or trapped.
- Often paired with candle rejection and structural context.

Why it ranks here

- Strength — Useful around extreme spikes and liquidity events.
- Strength — Adds information not identical to EMA trend logic.
- Risk — Countertrend entries can be dangerous during true expansion.
- Risk — Volume thresholds are session-dependent.
- Risk — Needs a hard regime block against strong trend continuation.

Bot alert procedure

ALERT ROUTE

Only alert the final confirmed reversal signal, never raw volume-spike detection. For bot use, require position-flat and a hard SL in the JSON.

Final Straw Confluence

Rank	Class	Score	Status
#15 Research	ORANGE	5.5/10	KEEP / TEST

Observed performance: No standardized performance snapshot preserved in current history.

Core logic

- Combines multiple trend, momentum, structure, candle, and volume confirmations into a high-confluence entry threshold.
- Designed to trade only when enough independent evidence agrees.

Why it ranks here

- Strength — Can produce very selective A+ setups.
- Strength — Useful concept for manual confidence scoring.

- Risk — Rule stacking can silently overfit.
- Risk — Multiple correlated filters may appear independent but measure the same move.
- Risk — Can end with too few trades to validate statistically.

Bot alert procedure

ALERT ROUTE

If used, trigger only the final confluence-approved strategy entry. Avoid separate alerts for each component.

GENIUS ONE V19.2 — Diagnostic Trade Engine

Rank	Class	Score	Status
#16 Diagnostic Only	RED	4.0/10	ARCHIVE / DIAGNOSTIC

Observed performance: Purpose was to force historical EMA-cross trades so Strategy Tester would populate.

Core logic

- Diagnostic mode uses simple EMA-cross style entries to prove strategy execution/reporting works.
- After Strategy Tester data appears, diagnostic mode is turned off to test the actual entry engine.

Why it ranks here

- Strength — Excellent troubleshooting tool when reports show no data.
- Strength — Separates Pine execution issues from strategy logic issues.
- Risk — Not a production edge.
- Risk — Any profit from diagnostic mode is irrelevant to final strategy quality.

Bot alert procedure

ALERT ROUTE

Never connect diagnostic mode to the live execution webhook. Keep alerts disabled or point them to a test/log endpoint only.

3. TradingView Alert Message & JSON Runbook

PRODUCTION DEFAULT: When a strategy order already carries a dynamic alert_message=..., create a TradingView **Order fills** alert and put exactly {{strategy.order.alert_message}} in the Message box. That placeholder tells TradingView to send the JSON/string generated by the script for the actual order that filled.

The decision rule: leave it, use the placeholder, or paste JSON?

What the Pine script does	TradingView alert condition	What to put in Message	Bot use
strategy.entry() / strategy.exit() contains alert_message= ...	Order fills / strategy order fills	{{strategy.order.alert_message}}	Preferred production route. The JSON is already built inside the script and changes with BUY/SELL, symbol, contracts, TP/SL, reason, etc.
Script calls alert(dynamicJSON, ...)	Any alert() function call	Do not replace the script-generated payload with a static JSON template. The alert() call supplies the message itself.	Use only if you intentionally want the early/intrabar route. Do not run it to the same execution webhook at the same time as fill alerts unless the bot de-duplicates.
Script uses alertcondition() or an indicator condition but does not attach order alert_message	Select the named BUY/SELL condition	Use the JSON template expected by the bot, with valid TradingView placeholders where appropriate.	Needed for indicator-only/legacy alert routes. The JSON is not automatically supplied by a strategy order.
JSON appears only inside input.text_area(), comments, notebook text, or instructions	Depends on the real alert code	Do not assume that JSON is active. It is only documentation unless the source passes it into alert() or an order's alert_message.	Trace where the JSON is actually used before connecting the bot.

Why {{strategy.order.alert_message}} matters

If the script has code similar to `strategy.entry("LONG", strategy.long, alert_message=strategy_long_exec_msg)`, the variable `strategy_long_exec_msg` is the real dynamic payload. TradingView does not need you to manually copy that JSON into the Create Alert window. The Message field should contain the placeholder `{{strategy.order.alert_message}}`, which is replaced at execution time with the exact message attached to that order.

Do not paste one fixed BUY JSON into an Order fills alert. A static message can be wrong when the next fill is SELL, a partial exit, TP2, SL, or CLOSE. The script's order-specific `alert_message` is safer because it can carry the correct side, quantity, reason, release/version tag, and other live values.

V15.x / V16 / V20 family - recommended bot alert setup

These versions use strategy orders and, in the versions shown in this runbook, also have separate early/intrabar `alert()` routing. For the production bot, the cleanest setup is the strategy-fill path:

1. In the script settings, use **FILLS ONLY** when that option exists, or disable executable EARLY/intrabar routing.
2. TradingView – **Create Alert** – Condition = the strategy – choose **Order fills / Order fills only**.
3. In **Message**, enter exactly `{{strategy.order.alert_message}}`.
4. Enter the bot webhook URL in the webhook field.
5. Create the alert. After changing Pine code, inputs that affect alert behavior, contract quantity, or JSON logic, recreate the alert so the running TradingView alert uses the intended strategy snapshot.

EARLY BUY/SELL labels are not the same thing as strategy fills. If you create a second alert on “Any alert() function call” and send it to the same execution endpoint, one setup can reach the bot twice: once early and once when the strategy order fills. Use a separate notification endpoint for early signals unless duplicate suppression is deliberately implemented.

V19.1B OGX BUY-\$ELL\$ - exact alert behavior

V19.1B is especially clear because it has two bot modes in the source: **Strategy Order Fills** and **alert() Immediate**. The functions `buildEntryJson()`, `buildCloseJson()`, and `buildPartialCloseJson()` create dynamic JSON inside the script.

V19.1B Bot Alert Mode	Create Alert condition	Message field	Recommendation
Strategy Order Fills	Strategy – Order fills	<code>{{strategy.order.alert_message}}</code>	Use this for production. Entries, closes,

V19.1B Bot Alert Mode	Create Alert condition	Message field	Recommendation
			partial closes, TP1/TP2/SL events can use the JSON attached to the actual strategy order.
alert() Immediate	Any alert() function call	The message is generated by sendBot(entryMsg)/alert(msg,...); do not substitute a fixed JSON payload.	Use only when intentionally trading the immediate alert path and after realtime testing.

The large **BOT ALERT TEMPLATE (WORKING)** inside the V19.1B input.text_area() is useful as documentation/manual fallback, but the script's automatic JSON builders are the stronger route. When **Strategy Order Fills** is selected, use {{strategy.order.alert_message}}; do not manually paste the example template into the alert box unless you deliberately bypass the auto JSON architecture.

How to tell whether JSON is really “already in the script”

Search the Pine source for these patterns. The first two mean the payload is executable; the third may only be documentation:

1) strategy.entry(..., alert_message = someJson) strategy.exit(..., alert_message = someJson) strategy.close(..., alert_message = someJson) 2) alert(someJson, alert.freq...) 3) input.text_area("{ ... JSON ... }") <- may be notes only unless referenced by #1 or #2

If #1 is present, the fill-alert Message is normally {{strategy.order.alert_message}}. If #2 is the route you chose, use **Any alert() function call**. If only #3 exists, you still need to determine how the actual alert is sent.

Production alert safety check

Check	Pass condition
One executable source per setup	Fill route OR immediate alert() route - not both to the same bot unless de-duplication is proven.
Correct Message field	Order fills with dynamic order JSON = {{strategy.order.alert_message}}.
Exit messages	Bot understands CLOSE/PARTIAL_CLOSE/TP/SL payloads before live use.

Check	Pass condition
Symbol and contract mapping	TradingView ticker maps to the correct Tradovate/broker contract and quantity.
Alert snapshot	Alert was recreated after material Pine/input changes.
Dry run	Webhook/log output verified before funded execution.

Important: Chart labels, dashboard text, notebook JSON, and alert code are separate things. A BUY label appearing on the chart does not prove that the bot received an executable BUY message. Verify the exact alert route.

4. Indicator & Filter Ranking

#	Indicator / Filter	Class	Score	Reason
1	Session / Time-of-Day Regime	FILTER	9.8/10	Separates high-quality NQ windows from dead/chop periods. Your own tests repeatedly showed time-of-day changes expectancy.
2	Market Regime Engine	FILTER	9.7/10	TRENDING / TRANSITION / CHOP / UNSTABLE classification controls when the same setup should or should not trade.
3	VWAP + Location	FILTER	9.6/10	Strong context filter: above/below VWAP, expansion from VWAP, overlap/compression, and mean-reversion

#	Indicator / Filter	Class	Score	Reason
4	EMA Structure / Slope / Spread	FILTER	9.5/10	distance. Fast directional context plus compression/expansion. EMA 5/13 and wider 21/48/200+ stacks recur across your strongest builds.
5	Volume + Impulse	FILTER	9.4/10	Confirms whether breakout/reversal has real participation; volume relative to MA plus ATR/range expansion is more useful than raw volume alone.
6	ATR / Volatility State	FILTER	9.2/10	Normalizes stops, expansion, compression, and late-entry detection across different NQ volatility regimes.
7	Trend Persistence	FILTER	9.1/10	Counts sustained closes above/below EMA/VWAP plus slope and directional closes; helps distinguish continuation from a one-bar spike.
8	MAE/MFE	FILTER	9.1/10	Directly

#	Indicator / Filter	Class	Score	Reason
	Excursion Logic			addresses the costliest failure pattern: entries that immediately move against the position or give back favorable excursion.
9	BOS / CHoCH / Pivot Structure	FILTER	8.9/10	Adds actual structural confirmation; best when pivots are confirmed and not repainting.
10	MACD / Histogram	FILTER	8.4/10	Good momentum timing and fade detection, but lagging and correlated with EMA trend.
11	Candle Body / Hammer / Wick Quality	FILTER	8.3/10	Useful trigger after location and regime are established; weak as a standalone signal.
12	Opening Range (NYC / Asia)	FILTER	8.2/10	Strong location framework for breakout/retest and liquidity behavior.
13	ADX	FILTER	7.8/10	Useful trend-strength filter, but slower and less directly actionable than regime + ATR + EMA slope.

#	Indicator / Filter	Class	Score	Reason
14	Compression Engine	FILTER	7.8/10	Good trade blocker when EMA/VWAP/ATR/volume all contract; can also help anticipate expansion.
15	Ghost Feed / Volume Anomaly	FILTER	7.0/10	Useful specialist signal; should not override strong directional regime.
16	Heikin-Ashi Candle State	FILTER	6.0/10	Good visualization/confirmation, poor execution truth source unless raw-price execution is preserved.

Preferred Confluence Stack

TIME + MARKET REGIME + LOCATION + STRUCTURE + TRIGGER

Example production logic: 09:30–11:30 ET + TRENDING/clean TRANSITION + above VWAP + structure breakout/retest + bullish impulse/candle trigger = LONG. The strongest versions work because they combine context first and trigger last, instead of letting one indicator create the trade.

5. TradingView → Bot Alert Runbook

PRIMARY RULE

Use ONE executable signal path per strategy. The most reliable default for automation is strategy ORDER FILLS using {{strategy.order.alert_message}}. Treat early/intrabar alert() signals as a separate mode, not an additional duplicate execution route.

Step	Procedure
1. Load one production strategy	Use a clean chart/layout with only the strategy that will control execution. Confirm symbol, contract, timeframe, and session.

Step	Procedure
2. Set script alert mode	For V15/V16/V20 families: prefer FILLS ONLY / strategy fills for production. For V19.1B: Bot Alert Mode = Strategy Order Fills.
3. Create TradingView alert	Open Create Alert. Under Condition choose the strategy, then choose Order fills (or the strategy option that includes order fills only).
4. Message	Set the Message field to <code>{{strategy.order.alert_message}}</code> when the script builds its own JSON in <code>strategy.entry/strategy.exit</code> .
5. Webhook	Enable Webhook URL and paste the bot's HTTPS endpoint. TradingView posts the alert body to that endpoint.
6. Prevent duplicates	Do not also send Any alert() function call to the same execution webhook unless the bot de-duplicates setup IDs and knows which path has priority.
7. Verify payload	Use a paper/test endpoint first. Confirm BUY/SELL/CLOSE, symbol, contracts, TP/SL, and split quantities are correct.
8. Verify state	One strategy = one bot execution stream. Confirm bot shows FLAT before enabling, then test one entry and one exit.
9. Forward test	Run realtime/paper before funded/live. Compare alert timestamps, chart fills, bot fills, slippage, partial exits, and re-entry behavior.
10. Archive alert settings	Record strategy version, chart timeframe, symbol, script inputs, alert mode, webhook endpoint label, and date created.

Alert Mode Matrix

Family	TV Condition	Message	Bot?	Notes
V20.4.6 / V20.3 / V16 CLEAN / V15.x	Order fills	<code>{{strategy.order.alert_message}}</code>	YES	Recommended fill-synchronized production route; V20.4.6 can alternatively run EARLY + INTRABAR as a separate

Family	TV Condition	Message	Bot?	Notes
V15.x EARLY mode	Any alert() function call	Script-generated JSON	Only if intentionally trading early	execution mode Higher intrabar/repaint /duplicate risk
V19.1B	Strategy Order Fills	{{strategy.order.alert_message}}	YES	Best use of its JSON builder
V19.1B alert() Immediate	Any alert() function call	Script-generated JSON	Test only / optional	Do not run with fill execution simultaneously
EMA+MACD / Trend Continuation	Order fills if strategy	{{strategy.order.alert_message}}	YES	Otherwise final BUY/SELL alertcondition
Indicator-only dashboards	Final confirmed BUY/SELL condition	Custom JSON	Conditional	Need bot-side position/cooldown controls
Diagnostic V19.2	None / test log endpoint	N/A	NO	Never connect diagnostic mode to funded execution

Bot-Safe JSON Minimum

MINIMUM FIELDS

```
{"action":"BUY|SELL|CLOSE","symbol":"MNQ...","contracts":2,"signal":"ENTRY|CLOSE","version":"Vxx","slPts":20,"tp1Pts":10,"tp2Pts":25}
```

- Prefer script-generated JSON over manually typed static JSON when position size, split, or target values can change.
- Include a version/release tag so the bot log shows exactly which strategy generated each command.
- Include an idempotency/setup key if you ever intentionally combine EARLY + FILL alerts; otherwise a network retry or dual route can duplicate orders.
- Never put account passwords, broker credentials, API secrets, or authentication tokens inside the TradingView alert body.

6. Production Validation Checklist

Gate	Pass Requirement
Backtest integrity	Raw candles; correct symbol/contract; realistic commission/slippage; same date range; no accidental HA execution.
Sample size	>500 trades preferred for broad conclusions;

Gate	Pass Requirement
	>2,000 trades is much stronger if regime coverage is broad.
Risk	PF >1.5 minimum research threshold; >2.0 strong. Max DD must fit funded-account buffer and daily loss rules.
Stability	Test 1m/2m/3m/5m separately; do not select a timeframe only because it is the single best historical result.
Session robustness	Break results into 03:00–06:30, 08:30–11:30, 12:00–13:30, 13:30–15:00, and evening windows.
Execution	Forward-test alert timestamps vs strategy fills; verify no duplicate BUY/SELL, partial exits, or same-bar flips.
Walk-forward	Optimize on one period, validate on a later untouched period. Reject settings that collapse out-of-sample.
Stress test	Add slippage, commission, delayed entry, and one-tick worse fills. Strong strategy should degrade gradually, not collapse.
Version control	Never overwrite V20.4.6, V20.3, V16 CLEAN, or V15.0 benchmark/control versions. Branch new experiments with a new version tag.

Recommended Deployment Tiers

Tier	Strategy	Operating Rule
Tier A — Conservative Funded	V20.4.6 FILLS ONLY, V20.3, or V15.1	Prioritize minimum drawdown; fills only; smaller size; no experimental early execution.
Tier B — Balanced Production	V16 CLEAN or V15.0	Higher throughput; fixed tested risk; fills only; strict daily loss stop.
Tier C — Research	V15.5 / V17 / V16.3	Paper/forward test; compare module attribution; no funded automation until isolated.
Tier D — Archive / Diagnostic	V19.1B / HA legacy / V19.2 Diagnostic	Keep useful components, but do not use current observed versions as primary funded bot.

7. Sources, Evidence & Limitations

- Uploaded source: MCURVE OGX / GENIUS ONE V15.0 Performance Recovery Edition (2,301 lines). This provided exact defaults and detailed V15 logic.
- Current conversation screenshots supplied performance snapshots for V15.0, V15.1, V16 CLEAN, V20.3, V20.4.6, and V19.1B.
- Current conversation pasted source supplied detailed V15.5 PF Quality and V19.1B logic.
- Historical strategy names/logic not fully pasted in this run were summarized from prior development notes. Those sections are marked as historical rather than presented as exact line-by-line source reconstruction.
- TradingView alert instructions were checked against current TradingView documentation on strategy alerts, alert() calls, and webhook alerts.

IMPORTANT COMPARISON LIMITATION

A ranking is only fully apples-to-apples when symbol, contract month, timeframe, date range, commission, slippage, order size, session settings, and strategy properties are identical. Treat the present ranking as an engineering decision matrix, then lock a standardized test harness for final selection.

TradingView references checked (Aug 2026):

- TradingView Help — Strategy Alerts
- TradingView Pine Script Docs — Alerts / alert_message
- TradingView Help — How to configure webhook alerts
- TradingView Help — Alerts on alert() function
- TradingView Help — Getting started with technical alerts