Parameter explanations

* Max position notional % of equity (100=full)
  + - Caps gross exposure at entry as a percent of realized equity at that moment.
    - Enforced after risk-based sizing.
    - Example: equity=10,000; max\_notional=50% → position value ≤ 5,000. If risk sizing suggests a larger position, it’s reduced to fit this cap.
* Equity cap for sizing (0=none)
  + - Caps the equity used for risk sizing to a fixed dollar amount: effective\_equity = min(realized\_equity, equity\_cap) when equity\_cap>0.
    - Example: equity=10,000, equity\_cap=5,000, risk=5% → risk\_cash = 5% of 5,000 = 250 (not 500).
* Time stop bars (0=off)
  + - Exits a trade after N bars at that bar’s close if neither stop nor take-profit hit earlier.
    - 0 disables this time-based exit.
* Single-position mode (y/N)
  + - When enabled: do not open a new trade if any trade is currently open (no overlaps).
    - When disabled (default): overlapping trades are allowed; each new trade uses the current realized equity at entry for sizing.

Notes on sizing and costs

* Risk % controls intended max loss via stop distance: risk\_cash = realized\_equity\_at\_entry × risk% → qty = floor(risk\_cash / stop\_distance).
* Max position cap and equity cap further constrain qty.
* Slippage and fees are applied to fills; they can make realized loss slightly exceed the nominal risk%—that’s expected unless you prefer sizing based on slippage-adjusted entry (I can switch to that if you want).

Option g runs the Backtest Markers flow. Here’s what will happen and what to enter.

* File selection: A picker opens. If you cancel, it uses C:\Projekte\crt\_250816\data\raw\btc\_1day\_candlesticks\_all.csv.
* Timespan: You’ll see the available range, then a date (or date+time) dialog:
  + - OK: uses your chosen start/end
    - Default: uses full span
    - It echoes the applied span and row counts, and the span is embedded in the XLSX filename
* Markers: It lists markers CSVs in results/. Pick one or enter 0 for ALL (e.g., all doe\_markers\_\*.csv). Markers are filtered to your span.
* Parameters (defaults used on empty):
  + - Risk per trade % (default 5)
    - Trailing stop % (default 5)
    - Take profit % (default 25)
    - Fee per side % (default from BT\_FEE\_PCT or 0)
    - Slippage % (default from BT\_SLIPPAGE\_PCT or 0)
    - Max position notional % of equity (default 5) caps gross exposure size
    - Equity cap for sizing (default 0 = none)
    - Time stop bars (default 0 = off)
    - Single-position mode (y/N)
* Output:
  + - Console summary of KPIs
    - XLSX report: results/backtest\_sYYYYMMDD[\_HHMM]-eYYYYMMDD[\_HHMM]\_TIMESTAMP.xlsx
    - Sheets: Summary (includes span and params), Trades (with Entry/Exit fills, Risk\_Cash, price audit columns), EquityCurve (dual-axis Price/Equity chart), DOE\_Perf and per-analysis DOE pivots if applicable

Tips

* To debug the time picker, set BT\_DEBUG\_TS=1 before running to see raw/parsed outputs.
* Global defaults: BT\_FEE\_PCT, BT\_SLIPPAGE\_PCT, BT\_MAX\_POS\_VAL\_PCT, BT\_EQUITY\_CAP, BT\_TIME\_STOP\_BARS, BT\_SINGLE\_POSITION.
* DOE: ProcessPool + timing
  + - Switched DOE parallelization to ProcessPoolExecutor for CPU-bound work.
    - Logs total DOE duration: “DOE completed in Xs across N parameter pairs.”
    - Keeps deterministic aggregation order and all existing outputs.
* DOE: Walk-forward CV + Robust score + Pareto overlay
  + - Optional via env vars:
    - DOE\_WF\_SPLITS: number of folds (e.g., 3). If >1, runs walk-forward CV per parameter pair and per divergence type.
    - DOE\_ROBUST\_LAMBDA: robust score = mean − λ·stdev (default 1.0).
    - DOE\_TOPN\_OVERLAY: overlay top-N cells by robust score (default 5).
    - Overlays:
    - “S:<score>” markers on top-N cells by robust score.
    - “P” markers on cells on the Pareto front (maximize mean, minimize stdev).
    - Applies to the 2×2 DOE by-type facet HTML.
* Backtester: conservative sizing option
  + - New prompt toggle: “Conservative sizing (use entry fill for stop distance)? (y/N)”
    - If enabled, sizing uses entry\_fill instead of entry\_price for the stop distance and may reduce qty; PnL recomputed if qty shrinks.
* Backtester: concurrent caps
  + - New prompts:
    - Max concurrent open positions (0=unlimited)
    - Max concurrent open LONGS (0=unlimited)
    - Max concurrent open SHORTS (0=unlimited)
    - Enforced before opening a new trade (in addition to single-position mode).
* Backtester: theoretical max loss audit
  + - Trades sheet includes ‘Theoretical\_Max\_Loss\_$’: stop-based worst-case (includes slippage and fees) at the stop fill.
* XLSX axes/titles
  + - Already enabled for EquityCurve and DOE Overview. Nothing else to do here.
    - Notes and usage
* DOE ProcessPool + timing
  + - No flags required. To control workers: DOE\_MAX\_WORKERS.
    - You will see a DOE timing line after the run completes.
* Robust DOE overlays
  + - Set DOE\_WF\_SPLITS=3 (or higher) to enable walk-forward CV.
    - Optional: DOE\_ROBUST\_LAMBDA=1.0, DOE\_TOPN\_OVERLAY=5.
    - Overlays appear on the 2×2 DOE by-type facet as white “x” markers with text (“S:<score>”, “P” for Pareto).
* Backtest prompts (updated)
  + - Conservative sizing toggle — on for stricter risk accounting.
    - Position caps — number of open positions globally and per-direction.
    - Existing sizing still uses realized equity at entry and honors max position notional % and equity cap.

Great ideas — I implemented the requested enhancements and made outputs include the asset/frequency tag.

What I added

* Asset/frequency tag in all outputs
  + Filenames: now prefixed with the asset tag (e.g., btc\_1day)
    - Markers CSV/XLSX: results/btc\_1day\_markers\_output\_….csv
    - HTML plots: btc\_1day\_plot.html, btc\_1day\_doe\_heatmap\_total.html, btc\_1day\_doe\_heatmaps\_by\_type.html, btc\_1day\_doe\_heatmap\_<Type>.html
    - DOE summary XLSX: btc\_1day\_doe\_summary\_….xlsx
    - Backtest XLSX: btc\_1day\_backtest[\_doe\_combined]*s…-e…*….xlsx
  + Inside files:
    - Markers CSV/XLSX: added column Asset\_Tag
    - DOE Summary XLSX: adds Info sheet with Asset\_Tag; DOE\_Summary also carries Asset\_Tag column
    - Backtest Summary XLSX: adds Asset\_Tag column in Summary
* Robust DOE Backtest (walk-forward CV on KPIs)
  + Enable by env vars (no menu change needed):
    - DOE\_BT\_WF\_SPLITS: number of folds > 1 enables CV
    - DOE\_BT\_ROBUST\_KPI: one of Total\_PnL, Win\_Rate\_%, Profit\_Factor (default: Total\_PnL)
    - DOE\_ROBUST\_LAMBDA: λ in robust score = mean − λ·std (default: 1.0)
  + Produces a DOE\_BT\_Robust sheet in the DOE XLSX with:
    - Candle\_Percent, MACD\_Percent, KPI\_mean, KPI\_std, Score
  + Overlays Top-N and Pareto markers on the 2×2 by-type DOE facet:
    - DOE\_TOPN\_OVERLAY controls N (default 5)
    - Labels show “BT S:<score>” and “P” for Pareto cells
  + Uses default backtest params (env defaults for fee/slippage). If you want specific backtest settings for this CV, we can add envs for risk/stop/tp too.
* DOE engine: ProcessPool + timing
  + Switched DOE parallelization to ProcessPoolExecutor for CPU-bound work.
  + Prints timing: “DOE completed in Xs across N parameter pairs.”
* Backtester: conservative sizing, concurrent caps, audit column
  + Conservative sizing: prompt toggle to use entry\_fill for stop distance (more conservative qty), recomputes PnL if qty shrinks.
  + Caps in prompt:
    - Max concurrent open positions (0=unlimited)
    - Max concurrent LONGS/SHORTS (0=unlimited)
  + Added Theoretical\_Max\_Loss\_$ per trade (includes fees/slippage at stop fill).
* XLSX chart axes
  + Overview charts now include x-axis titles (Analysis, MACD tolerance, Candle tolerance).
  + EquityCurve already shows X (Date), Y1 (Price), Y2 (Equity), with title and legend.

Notes and usage

* Asset tag extraction
  + From the input filename, e.g., btc\_1day\_candlesticks\_all.csv → btc\_1day
  + Used in titles, filenames, and Info/columns as described.
* Robust DOE Backtest CV
  + - Set:
    - DOE\_BT\_WF\_SPLITS=3 (or more)
    - Optional: DOE\_BT\_ROBUST\_KPI=Total\_PnL|Win\_Rate\_%|Profit\_Factor
    - Optional: DOE\_ROBUST\_LAMBDA=1.0
    - Optional: DOE\_TOPN\_OVERLAY=5
  + Runs CV per param cell using in-memory markers and backtesting on each fold.
  + Adds DOE\_BT\_Robust sheet and overlays robust markers onto the DOE facet HTML.
* DOE Robust (marker counts) CV still supported
  + Enable via DOE\_WF\_SPLITS>1; overlays “S:” and “P” based on counts.
  + If both BT and counts robust overlays are enabled, the BT overlay takes precedence.

Below is a practical blueprint to find robust, high‑performing parameter sets across assets and timeframes, while controlling overfit and drawdowns.

Goals

* Primary (indicator): candle\_percent, macd\_percent
* Secondary (trading): risk\_pct, trailing\_stop\_pct, take\_profit\_pct
* Robust across assets and frequencies (1h/4h/1d/1w), maximizing return while minimizing loss/drawdown.

Method: staged, robust, multi-objective

1. Stage A — Wide DOE on indicators (fast, counts + coarse PnL)

* Grid sample of candle% × macd% per asset × timeframe.
* Use current DOE + “Robust” overlays:
  + Marker-count CV: mean − λ·std on Total markers (cheap prefilter).
  + Backtest CV (Opt-in): mean − λ·std on a KPI (e.g. Total\_PnL, PF, WinRate).
* Keep only plateau cells (top-N robust by score and Pareto front).
* Output: short list of promising indicator cells per asset/frequency.

1. Stage B — Secondary sweep on trading parameters (focused)

* For each shortlisted indicator cell:
  + Sweep risk\_pct, trailing\_stop\_pct, take\_profit\_pct (coarse grid).
  + KPI set: Total\_PnL, Profit\_Factor, WinRate, MaxDD, Expectancy, Trades count.
  + Robust score: mean(KPI) − λσ(KPI) over walk‑forward splits (see CV).
  + Constraints: min Trades, MaxDD ≤ threshold, PF ≥ threshold, WinRate ≥ threshold.
* Produce Pareto set (maximize Total\_PnL/PF; minimize MaxDD/volatility).

1. Stage C — Bayesian fine‑tuning (local)

* For best few (indicator+trading) candidates, run Bayesian optimization (e.g., Optuna/TPE or skopt) around those neighborhoods.
* Objective can be:
  + Scalarized: Score = α·Z(PnL) + β·Z(PF) − γ·Z(MaxDD) with α,β,γ set; or
  + Quantile focus: maximize PnL subject to PF and DD constraints; or
  + Robust: mean − λ·std across splits.
* Early stop if no improvement after K iterations.

1. Stage D — Walk‑Forward Validation (WFO) as standard

* Purged, expanding windows: split the history into K folds in chronological order:
  + Train on [start…t\_k], validate on (t\_k…t\_k+1].
  + Aggregate KPIs: mean, std, 5th/95th quantiles; compute robust score = mean − λ·std.
* Optional Monte‑Carlo: bootstrap resample fold‑level returns or run path shuffles to estimate edge stability.

1. Stage E — Out‑of‑sample and forward monitoring

* Fix the chosen parameter set(s) and run out-of-sample (last period not used in any tuning).
* Log forward KPIs and alert if performance drifts beyond tolerance.

Design choices that keep you safe

* Minimum trades filter per fold (e.g., ≥ 30) to avoid spurious wins.
* DD guardrails (e.g., MaxDD ≤ 30%).
* Risk‑of‑Ruin estimate: penalize score if RoR above threshold for assumed Kelly fraction.
* Fee/slippage included (yours already are), and scenario tests: stress fees/slippage ×2 and re-evaluate robustness.
* Conservative sizing option (use entry\_fill distance) ON for tuning to avoid optimistic sizing bias.

Concrete parameter ranges (start coarse, then refine)

* Indicator
  + candle\_percent: 0.05–1.5 (daily); adjust tighter on intraday (e.g., 0.02–1.0)
  + macd\_percent: 0.5–10 (daily); adjust per timeframe (e.g., 0.3–7 intraday)
* Trading
  + risk\_pct: 0.5–5% (start 0.5, 1, 2, 3, 4, 5)
  + trailing\_stop\_pct: 2–15% (intraday: 1–10)
  + take\_profit\_pct: 5–50% (intraday: 3–30)
* Timeframe scaling
  + Lower TF → narrower stop/tp ranges and smaller risk\_pct
  + Higher TF → wider stop/tp and possibly higher risk\_pct cap

Multi-objective scoring

* Scalarized option:
  + Score = Z(PnL) + 0.5·Z(PF) − 0.5·Z(MaxDD)
* Robust option (default):
  + Score = mean(PnL) − λ·std(PnL) with λ ∈ [0.5, 2]; filter PF≥1.3, MaxDD≤30%
* Pareto overlay: annotate DOE and secondary sweeps with nondominated points on (mean, std) and (PnL, MaxDD).

How to implement with your current code quickly

* Already available
  + DOE ProcessPool with overlays on counts and backtest CV (env toggles).
  + Backtester with conservative sizing, concurrency caps, equity audit columns.
* Next steps (minimal code deltas)
  + Add a “secondary sweep” function:
    - Input: list of indicator cells, grids for risk/stop/tp
    - For each tuple: run walk‑forward backtest, compute KPIs and robust score
    - Output: XLSX “Secondary\_Sweep” with tables + Pareto plots per asset/frequency; overlay top-N on a scatter (PnL vs MaxDD).
  + Add CLI/ENV controls:
    - SECONDARY\_GRID\_RISK="0.5,1,2,3,4,5"
    - SECONDARY\_GRID\_STOP="2,4,6,8,10,12,15"
    - SECONDARY\_GRID\_TP="5,10,15,20,25,30,40,50"
    - SECONDARY\_WF\_SPLITS (default 3)
    - SECONDARY\_SCORE="robust" or "scalar"
  + Leverage your existing walk‑forward and ProcessPool tooling; write results into a dedicated XLSX with an Overview and per-cell tabs.

Cross‑asset generalization

* Per‑timeframe defaults: tune on a diverse asset set; pick robust parameters common across assets in same timeframe (plateau intersection).
* Meta‑learner: as a later step, learn mapping from asset features (volatility, ATR/price) and timeframe to suggested parameters; use that for cold‑start.

Validation checklist

* Include fees and slippage in tuning (done).
* Use conservative sizing during tuning; you can relax later but keep guardrails.
* Enforce min trades per fold and KPI floors.
* Export full audit in XLSX: per-fold metrics, summary with robust score, and overlays.

Here are concrete CSV examples you can drop into secondary\_sweep\_ranges.csv to override the “all” defaults per frequency. Important: use one schema per file (list form or range form), not both in the same CSV. The optimizer gives precedence CSV > ENV > built‑ins. Frequency is parsed from the asset tag’s second token (e.g., btc\_1h, eth\_4h, sol\_1day, btc\_1week).

List form (same header as your current file)

* Schema: param,values,frequency
* values: comma/semicolon‑separated numbers

Example (add below your existing “all” rows)  
param,values,frequency  
candle\_percent,"0.005,0.01,0.02,0.05,0.10,0.20",1h  
macd\_percent,"0.5,1,2,3",1h  
risk\_pct,"1,2,3,5,8",1h  
stop\_pct,"1,2,3,5,7",1h  
tp\_pct,"2,3,5,7,10",1h  
candle\_percent,"0.01,0.02,0.05,0.10,0.20,0.30",4h  
macd\_percent,"1,2,3,4",4h  
risk\_pct,"2,3,5,8,12",4h  
stop\_pct,"2,3,5,7,9",4h  
tp\_pct,"3,5,7,10,15",4h  
candle\_percent,"0.01,0.02,0.05,0.10,0.20,0.50",1day  
macd\_percent,"1,2,3,4,5",1day  
risk\_pct,"3,5,8,12,18,25",1day  
stop\_pct,"2,3,5,7,10",1day  
tp\_pct,"3,5,7,10,15,20,30,50",1day  
candle\_percent,"0.02,0.05,0.10,0.20,0.50,1.00",1week  
macd\_percent,"1,2,3,4,6,8,10",1week  
risk\_pct,"3,5,8,12,15",1week  
stop\_pct,"3,5,7,10,15",1week  
tp\_pct,"5,10,15,20,30,50",1week

Range form (alternative schema)

* Use this in a separate CSV file (or replace current one)
* Schema: param,min,max,step,frequency
* step can be number or the string auto (auto = ~6 points; log‑spaced for candle\_percent, linear for others)

Example (range form)  
param,min,max,step,frequency  
candle\_percent,0.005,0.20,auto,1h  
macd\_percent,0.5,3.0,0.5,1h  
risk\_pct,1,8,auto,1h  
stop\_pct,1,7,auto,1h  
tp\_pct,2,10,auto,1h  
candle\_percent,0.01,0.30,auto,4h  
macd\_percent,1,4,1,4h  
risk\_pct,2,12,auto,4h  
stop\_pct,2,9,auto,4h  
tp\_pct,3,15,auto,4h  
candle\_percent,0.01,0.50,auto,1day  
macd\_percent,1,5,1,1day  
risk\_pct,3,25,auto,1day  
stop\_pct,2,10,auto,1day  
tp\_pct,3,50,auto,1day  
candle\_percent,0.02,1.00,auto,1week  
macd\_percent,1,10,auto,1week  
risk\_pct,3,15,auto,1week  
stop\_pct,3,15,auto,1week  
tp\_pct,5,50,auto,1week

How it selects ranges

* The optimizer reads SS\_RANGES\_CSV (defaults to secondary\_sweep\_ranges.csv).
* It selects rows where frequency equals the asset’s frequency (1h/4h/1day/1week). If none match, it falls back to all rows. Then falls back to ENV, then built‑ins.
* For indicator candidates:
  + If CSV provides candle\_percent or macd\_percent lists/ranges, it builds the indicator grid from CSV.
  + Else, it uses doe\_parameters\_example.csv; if missing/invalid, it falls back to a small built‑in indicator set.

Tips

* Keep grids reasonable: total tasks ≈ |candle×macd| × |risk×stop×tp| × folds. Use coarse‑to‑fine (enabled by default) to keep runtime sensible.
* Verify asset tags follow <symbol>\_<frequency> (e.g., btc\_1h, eth\_4h, btc\_1day, btc\_1week) so frequency overrides match correctly.
* You can point the optimizer to a different CSV with SS\_RANGES\_CSV=path\to\your.csv.
* Thanks — I checked and tightened option m, and I’ve added clear “Voraussetzungen” (prerequisites) to the HTML/XLSX for all j–p options so it’s obvious when something needs to be run first.
* What I fixed or added
* Option m (Risk Management)
  + Now slices markers to your selected timespan when you pick the markers CSV (same as g/h). This prevents “no trades” due to out‑of‑span signals.
  + Computes simple diagnostics if trades exist (Win‑Rate, Avg Win/Loss, Payoff R, Kelly surrogate, Ruin proxy). Shows them in HTML and writes them to the Info sheet.
  + HTML includes “Schätzwerte” table and German bullets:
    - Voraussetzungen: marker CSV (from a–e or DOE f)
    - Was/Warum/Leer? and caveats
  + Always shows an equity chart for context.
* Option p (Portfolio)
  + If the number of overlapping bars across assets is tiny, many portfolios look identical. HTML now shows “Überlappende Zeilen” and explains the empty/identical case.
  + Even without overlap, it shows per‑asset Final\_Equity and PnL\_% individually so you still see differences.
  + Keeps at least 4 discrete weight choices per asset {0.10, 0.25, 0.50, 0.75}, samples 500 combos, and ranks best allocations (PnL\_% and Sharpe surrogate).
  + Adds correlation heatmap PNG, top allocations table, and a “Best Portfolio Equity” chart.
  + XLSX now adds sheets Asset\_Perf and Allocations.
* All new options j–p: prerequisites
  + j (WFCV), k (Execution), l (PBO/DS): bullets state you need marker CSV in results (from a–e markers export or DOE f) and to select timespan.
  + m: same, plus caveats on estimates.
  + n: requires date and close columns (enough data).
  + o: requires date column (regular frequency helps).
  + p: requires multiple asset files with date/close and sufficient overlap.
* Options j, k, l: richer visuals and explanations
  + Equity charts embedded as inline PNGs.
  + j: fold PnL bar chart; German bullets explaining fold/method/usage.
  + k: execution parameters and filtered markers shown; German bullets explain what/why.
  + l: adds German bullets about folds and purpose; OOS PnL bar chart; English method retained.
* Why m may still be empty
* Even after g/h, if markers within the chosen span do not produce trades under default backtest parameters (risk/stop/tp), you’ll see empty diagnostics. The module now tells you why and suggests generating markers first and/or adjusting backtest parameters in g/h.
* Quick usage tips
* When running j/k/l/m, pick the markers CSV and then select a timespan; markers are filtered to the span (echoed in the console).
* For p, ensure files share a common time window; otherwise, the joint portfolio returns will be degenerate (but you’ll still see per‑asset results).
* If you want, I can:
* Add console one‑liners for m–p (e.g., count of trades, top allocation PnL).
* Expose p’s weight choices and sample size via env vars (e.g., PORTF\_WEIGHTS, PORTF\_SAMPLES).