

Walk-Forward Factor Selection with Meta-Selection

Methodology for the JKP Characteristics Trading Factor (CTF) challenge · baseline model, submitted unmodified · Python, numpy/pandas/scipy only, fully deterministic

In one paragraph. Every characteristic in the CTF dataset is turned into a monthly long–short rank portfolio. A fixed, symmetric menu of 36 ways of choosing and weighting those factors is simulated walk-forward, so each way has its own genuine out-of-sample track record. Every six months a meta layer blends the three menu items with the best out-of-sample Sharpe ratio so far – so even the *methodology* in force at any date is chosen only from information available at that date. The blended factor weights are mapped to stock weights each month using that month's characteristics.

1. Data and timing convention

Input is the official CTF table `contrib_global_factor.ctff_chars` (1,369,324 stock-months, 864 month-ends from 1951-12 to 2023-11, 402 characteristics listed in `ctff_features`). `daily_ret` is accepted by `main()` but not used.

The row for stock i at month-end t holds characteristics known at the end of month t and the target `ret_exc_lead1m`, the excess return over month $t+1$. That target is realized, and therefore **known, only at the end of month $t+1$** . It is never used as a characteristic (it is not among the 402 features). The output weight w at month t is held over month $t+1$ and earns row t 's `ret_exc_lead1m`.

At the end of month...	What is known	What is decided
t	characteristics of month t ; factor returns of rows $\leq t-1$ (realized in months $\leq t$)	stock weights $w_{i,t}$ (ranks of month t); at a rebalance, factor selection, factor weights and the meta blend
$t+1$	<code>ret_exc_lead1m</code> of row $t \Rightarrow$ factor return row t	row t may now enter any later decision

2. The four stages

Stage 1 – Factor portfolio returns

For each characteristic k and month t , stocks are percentile-ranked cross-sectionally (missing values get zero weight) and weighted by `rank - 0.5`:

$$LS_{k,t} = \sum_i (r_{ik,t} - \frac{1}{2}) \cdot \text{ret}_{i,t+1} / \sum_i |r_{ik,t} - \frac{1}{2}|$$

A month needs ≥ 20 stocks with a valid rank, otherwise the value is missing; factors with fewer than 60 valid months are dropped. The result is an 864×402 matrix of long–short returns (unit gross exposure, approximately dollar-neutral).

No look-ahead. Ranks use only month- t characteristics. `ret_exc_lead1m` appears only as the *payoff* of the portfolio formed at t . The value stored in row t is treated by every later stage as unknown until the end of month $t+1$.

Stage 2 – Walk-forward simulation of a configuration menu

The menu is a full cross of 3 windows (10 years, 20 years, expanding) $\times$ 2 sizes (5 or 10 factors) $\times$ 2 selection rules $\times$ 3 weighting rules = **36 configurations**, fixed before looking at any result.

- **Sharpe**: the top- N factors by annualized Sharpe ratio over the window.
- **ClusterSharpe**: hierarchical clustering on $1 - |\text{corr}|$ of factor returns (average linkage, $\max(3N, 20)$ clusters); the best-Sharpe factor of each of the best N clusters. It avoids holding five versions of one factor.
- **EW** equal weights; **RankSharpe** weights proportional to Sharpe rank; **MeanVar** $w \propto \Sigma^{-1}\mu$ with 30% shrinkage of Σ toward its scaled identity, clipped at 0, normalized to sum to one, falling back to EW if degenerate.

Every six months (rebalance date rp) each configuration selects and weights factors, then earns the realized returns of those factors until the next rebalance. This gives every configuration its own out-of-sample (OOS) return series. Rebalances with under 24 months of history are skipped; 140 rebalances result. Factor selections shared by configurations that differ only in weighting are computed once and cached.

No look-ahead. At rp the selection and weighting functions receive only `ls_returns.loc[:rp - 1 month]`, i.e. rows realized by the end of month rp . Rolling windows end at that same row. The factor returns of the holding block (rows $\geq rp$) are used only to *record* what the configuration earned, never to choose it.

Stage 3 – Meta-selection

At each rebalance, each configuration is scored by the annualized Sharpe ratio of its own OOS series up to $rp - 1$ month (expanding). Among configurations with ≥ 120 months of OOS history, the top 3 are blended equally. Before any configuration qualifies, all active configurations are blended equally (an uninformative prior). In the full run, 120 of 140 rebalances were meta-selected; the first 20 used the equal-weight prior.

No look-ahead – and no configuration-selection bias. The OOS history used at rp contains only returns realized by the end of month rp . Because the methodology in force at any date is itself chosen from past OOS performance, the post-1990 record is not flattered by picking the best of 36 configurations with hindsight.

Stage 4 – Stock-level weights (monthly)

For each month t , the governing rebalance is the latest $rp \leq t$. For each of its chosen configurations and each factor k with weight fw_k , stock i receives

$$w_{i,t} = (1/K) \sum_{\text{configs}} \sum_k fw_k \cdot (r_{ik,t} - 1/2) / \sum_j |r_{jk,t} - 1/2|$$

so that each configuration's stock book exactly replicates its factor-level return. Weights are recomputed **every month** from that month's ranks (monthly rebalancing, as the rules require); the factor mix changes every six months. All non-zero weights are output as `(id, eom, w)`.

No look-ahead. Ranks use only month- t characteristics of stocks present at t ; the factor mix comes from a rebalance made at or before t . `ret_exc_lead1m` is not read.

3. Design constants fixed a priori

Constant	Value	Role
Rebalance / meta frequency	6 months	factor mix and meta blend update
Meta blend size	top 3	configurations blended equally
Meta minimum history	120 months	OOS months before a configuration can be chosen
Factor minimum history	60 months	valid months for a factor to be usable
Minimum stocks	20	valid ranks per factor-month
First rebalance	24 months	history required
Covariance shrinkage	30%	MeanVar weighting
Cluster multiplier	3×N (min 20)	ClusterSharpe

For small datasets (< 200 months, e.g. the 123-month CTF validation file) the minimum-history thresholds scale down (factor history $\max(6, n/5)$, 12-month first rebalance, meta history $\max(24, n/3)$) so the model still runs end to end.

4. Results (gross of trading costs)

Period (formation month)	Months	Annualized Sharpe (gross)	Ann. volatility at unit gross leverage
1960–1989	360	2.495	2.78%
1990–2003	168	3.009	3.47%
2004–2013	120	0.651	2.71%
2014–2023	119	0.656	2.77%
1990–2023 ($\approx$ CTF test period)	407	1.638	3.30%

Average monthly one-way turnover is **0.171** of the gross book. Mean gross exposure $\sum |w|$ is 0.41 (range 0.32–0.63) because long and short legs of different factors partly offset on the same stock; net exposure is essentially zero ($|\sum w| \leq 0.008$). All figures are gross of trading costs; a simple net adjustment is $\text{net}_t = \text{gross}_t - 2 \cdot \text{turnover}_t \cdot \text{cost}$ per side.

The meta layer's first informed choice is at 1963-12, when configurations first reach 120 OOS months. From then on it always blends 10-factor *ClusterSharpe* configurations, mostly with *MeanVar* weighting (6 of the 36 menu items are ever chosen). The sharp decline after 2004 is consistent with the documented post-publication decay of anomaly returns; because the model is honest out-of-sample, no attempt was made to tune it away.

5. Verification

Check	What was done	Outcome
V1 shift	Factor-return matrix shifted one month later before Stages 2–3 (one month staler information). 1990–2023 Sharpe 1.638 → 1.706; sub-periods move by –0.19 to +0.21.	PASS – only mild change
V2 truncation	Full pipeline run on data with eom < 1990-01-01 and on the full sample. Compared all 402 factor series, 2,628 factor selections and weight vectors (73 rebalances × 36 configurations), 73 meta-choices and 477,697 stock weights.	PASS – zero differences; stock weights bit-identical (max Δw = 0)
V3 replication	$\sum_i w_i \text{ret}_i$ versus the equally blended factor-level return of the chosen configurations, all 840 months.	PASS – correlation 1.000000, max diff 3.5×10^{-18}
V4 sanity	Annualized volatility at unit gross leverage and Sharpe by period.	PASS – vol 2.7–3.5%; max Sharpe 3.01 < 4
V5 determinism	Three separate processes (template, submission script, verification harness) wrote <code>weights.csv</code> .	PASS – byte-identical (identical SHA-256)
CTF validation	<code>main()</code> on 123-month subsets (all stocks; one stock in ten).	PASS – valid output (all six schema checks) on both subsets; weights for 105 of 123 months

V2 mirrors the challenge's automated look-ahead test: predictions on truncated data must equal those on the full data.

6. Every use of the forward return

Because `ret_exc_lead1m` is the single most common source of look-ahead, every line that touches it – directly, or through the factor-return matrix built from it – was reviewed. Line numbers refer to `strategy.py` (identical to `04-StrategyTemplate.py` except the last row, which moves down 3 lines because of the seed line).

Line(s)	Code	Why it is legitimate
93	<code>chars[["eom", "ret_exc_lead1m"] + valid_features]</code>	Stage 1 input. The target is kept as a separate column; it is not one of the 402 features.
94	<code>df[df["ret_exc_lead1m"].notna()]</code>	The factor portfolio at t is formed from stocks with a realized $t+1$ return; the resulting value is only used once it is realized (end of $t+1$). In this dataset no row is missing it, so the filter is a no-op. Stage 4 does not apply it.
104–110	<code>ret = month_data["ret_exc_lead1m"]; num = (ret * w).sum()</code>	The payoff of the portfolio formed with month- t ranks. Stored in row t , treated as known only at end of $t+1$.
119	<code>ls_df.notna().sum() >= min_obs</code>	Drops factors with < 60 valid months over the sample. It counts data <i>availability</i> , not returns; V2 shows it changes no pre-1990 decision (all 402 factors survive in both runs).
278–279	<code>sel_end = rp - 1 month; ls_returns.loc[:sel_end]</code>	Only rows realized by the end of month rp enter selection and weighting.
287–312	<code>threshold, window start, SELECT_FN, WEIGHT_FN</code>	All operate on <code>ls_avail</code> (rows $\leq rp - 1$), never on later rows.

314–315	<code>oos.loc[block] = ls_returns.loc[block, factors] @ weights</code>	Rows $\geq rp$ are the <i>outcome</i> of a decision already made; they are recorded as OOS P&L, not used to make it.
340	<code>hist = oos.loc[:rp - 1 month]</code>	Meta-selection sees only OOS returns realized by the end of month rp .
497–499	<code>weights.merge(... ret_exc_lead1m ...)</code>	Evaluation only, in the stand-alone runner after <code>main()</code> returns: weights at t earn row t 's forward return.

No unjustified use was found. One conservative quirk is documented but deliberately left unchanged: rebalances fall in June and December, and for June dates `rp - DateOffset(months=1)` gives May 30 rather than May 31, so June decisions exclude the May row and use information one month staler than permitted. This can only delay information, never leak it.

7. Reproducibility and limitations

- **Runtime:** about 5 minutes for the full sample on a 15.5 GB laptop (Stage 1 $\approx$ 2 min, Stage 2 $\approx$ 1.5 min).
- **Determinism:** no randomness is used; `np.random.seed(42)` is set at the start of `main()` for rule compliance only. Configuration and factor orders are fixed lists; ties in the meta ranking are broken by menu order.
- **Environment:** numpy 2.5.3, pandas 3.0.6, scipy 1.18.1, pyarrow 25.0.1 (pinned in `requirements.txt`; all have Python 3.13 wheels; pip-audit reports no known vulnerabilities).
- **Output:** `main()` returns (id, eom, w) for 1,361,631 stock-months over 840 months (1953-12 to 2023-11), with no missing values. `weights.csv` exports all rows with w written to 12 significant digits (max relative rounding error 5×10^{-12}) to stay under the 50 MB limit.
- **Limitations:** returns are gross of costs; the factor mix changes only every six months while stock weights are refreshed monthly; the menu and all constants were fixed a priori and not tuned on the test period, so no attempt was made to improve the weaker post-2004 performance.