

Equity Cashflow Projection Model

Full Technical Reference (40-Page Equivalent)

Version: 1.0

Scope: Current implementation in this repository

Primary code base: `src/ecf_model/` and `scripts/run_quarter_pipeline.py`

Math notation: LaTeX-style Markdown using \dots and \dots

Table Of Contents

1. Purpose, Scope, And Design Intent
2. System Architecture Overview
3. Data Lineage: From SQL Extracts To Model Inputs
4. Canonical Schema And Field Dictionary
5. Time Conventions, Quarter Logic, And Cutoff Behavior
6. Portfolio Snapshot Ingestion And Canonical Conversion
7. Fund Lifecycle Modeling (Investment End, Fund End, Overrun)
8. Base Feature Construction For Fitting
9. Timing Probability Model (Event Frequency)
10. Ratio Marginal Model (Cashflow Size)
11. NAV Dynamics Model (Omega Regression)
12. MSCI Model (Regime + AR Structure)
13. One-Factor Copula Dependence
14. Holdout Recalibration Framework
15. Stability Attenuation Framework
16. Group-Level Recalibration By Strategy And Grade
17. Phase-Specific Multipliers: Full Methodology
18. Simulation Engine: Quarter-By-Quarter Mechanics
19. Scenario Handling (Neutral / Bullish / Bearish)
20. Neutral Tuning Pipeline
21. Manual Timing Overlay (Timing Profile V2)
22. Output Artifacts And Interpretation
23. Diagnostics And Model Review Checklist
24. Statistical Defensibility: What Is Learned Vs Imposed
25. Risk, Limitations, And Failure Modes
26. Operational Runbooks (Windows / Mac / Linux)
27. Troubleshooting Guide
28. Governance, Change Control, And Documentation Standards
29. Suggested Enhancements
30. Appendix A: Formula Reference
31. Appendix B: Artifact Reference
32. Appendix C: Worked Quarter Example
33. Appendix D: Practical Calibration Interpretation
34. Appendix E: FAQ

1. Purpose, Scope, And Design Intent

The model projects private-market portfolio cashflows and NAV over quarterly horizons with a structure that balances:

- Statistical fit to historical behavior
- Practical robustness under sparse/noisy private market data
- Business realism at lifecycle boundaries (especially near and after fund end of life)

The model is designed to answer operationally important questions such as:

- What are expected cumulative drawdowns and repayments by quarter?
- What is expected end NAV and how quickly does NAV run off?
- What are resulting DPI, TVPI, and pooled IRR trajectories?
- How does behavior change under neutral, bullish, and bearish market assumptions?

The model is not a purely unconstrained machine learning black box. It is a transparent, semi-parametric system where each block can be audited and diagnosed.

2. System Architecture Overview

At a high level, the system has five layers:

1. Data ingestion and normalization
2. Base fitting on historical data up to cutoff
3. Holdout recalibration and lifecycle adjustments
4. Monte Carlo simulation with scenario paths
5. Optional neutral tuning against historical targets

Primary orchestration paths:

- Fit + project core: `src/ecf_model/pipeline.py`
- Neutral tuning: `src/ecf_model/tuning.py`
- End-to-end quarter run: `scripts/run_quarter_pipeline.py`

This separation is deliberate:

- Fit/calibration artifacts are portable and reusable.
 - Projection can be run on many test portfolios using a single calibration folder.
 - Tuning can be performed once and then reused for recurring runs.
-

3. Data Lineage: From SQL Extracts To Model Inputs

3.1 Business workflow

Typical data update path in this repository:

1. `data.ipynb`
2. `data_preparation.ipynb`
3. external anonymization step to produce `anonymized.csv`
4. update `msci.xlsx`

3.2 Model-facing inputs

Required operational inputs for modeling:

- Historical fund-level quarterly table: typically `anonymized.csv`
- MSCI history table: `msci.xlsx`
- Portfolio snapshot/history input for projection:
 - Canonical format OR
 - Snapshot template auto-converted by quarter runner

3.3 Why lineage matters

Most projection disagreement in production is caused by one of:

- Misaligned cutoff quarter across files
- Input file not in canonical shape
- Wrong portfolio snapshot date
- Non-comparable treatment of recallables

Hence, the pipeline aggressively canonicalizes schema and time fields.

4. Canonical Schema And Field Dictionary

The canonical requirements are implemented in `src/ecf_model/schema.py`.

Required columns:

- FundID
- Adj Strategy
- Grade
- Year
- Quarter
- First Closing Date
- Planned End Date
- Commitment EUR
- Adj Drawdown EUR
- Adj Repayment EUR
- Recallable

- NAV Adjusted EUR

4.1 Field semantics

- **FundID**: unique fund identifier
- **Adj Strategy**: normalized strategy label used for grouping
- **Grade**: normalized grade bucket used in calibration and tuning
- **Commitment EUR**: committed capital used as principal scaling anchor
- **Adj Drawdown EUR**: drawdown flow (signed or absolute in source; canonical logic takes absolute where needed)
- **Adj Repayment EUR**: repayment flow
- **Recallable**: recallable quantity (flow or cumulative depending on source)
- **NAV Adjusted EUR**: quarter-end adjusted NAV level

4.2 Canonicalization behavior

The loader (`src/ecf_model/data_io.py`) does:

- column rename mapping to canonical names
- type coercion to numeric/datetime as needed
- quarter-end timestamp construction
- strategy/grade text cleanup
- row filtering for valid fund and time

4.3 Recallable normalization

Recallable can arrive as either:

- period flow
- cumulative stock-like value

`recallable_to_flow` in `src/ecf_model/utils.py` detects monotonic cumulative series and converts via first differences (floored at 0). Otherwise, it treats as flow magnitude.

This is critical because recallable behavior affects both fitted distributions and simulation state transitions.

5. Time Conventions, Quarter Logic, And Cutoff Behavior

5.1 Quarter parsing

Quarter labels are standardized as `YYYYQn` and converted to quarter-end timestamps.

5.2 Cutoff semantics

Fit stage always uses only historical rows where:

$quarter_{end} \leq cutoff_{quarter}$

Projection start is set from the latest quarter in portfolio input.

5.3 Horizon determination

Projection horizon is not fixed calendar length by default. It is calculated from:

- start quarter
- model-implied end of life for active funds
- extra runoff buffer

This is why long-tail NAV and repayments can continue well beyond nominal horizon settings.

6. Portfolio Snapshot Ingestion And Canonical Conversion

The quarter runner (`scripts/run_quarter_pipeline.py`) supports two input types:

- canonical model format
- snapshot template with business-facing columns

When snapshot template is detected, it creates canonical CSV in:

`inputs/<cutoff>/...`

6.1 Template assumptions

Expected snapshot columns include:

- fund name
- strategy
- grade
- first closing date
- commitment amount

Optional draw/repay/nav fields are accepted.

6.2 Derived fields

The converter derives:

- quarter from cutoff
- age in quarters from first close to cutoff
- planned end date from first close + planned life assumption

This keeps runs reproducible across environments.

7. Fund Lifecycle Modeling

Lifecycle functions are in `src/ecf_model/features.py`.

7.1 Investment period end

`compute_invest_end_by_fund` uses:

- first close date
- first repayment timing

If repayments begin within 5 years, investment period is set shorter than for later-starting repayment profiles.

7.2 Fund end date

`compute_fund_end_dates` blends:

- planned end date
- historical strategy/grade overrun
- shrinkage toward broader levels
- fallback planned life when planned end is missing
- minimum life floor

Result: `fund_end_qe` per fund, used throughout phase logic.

7.3 Why this matters

All phase multipliers and tail behavior are keyed off `q_to_end` or `after_end_q`.

If fund end estimation is poor, timing mechanics can be directionally wrong even when base timing probabilities are accurate.

8. Base Feature Construction For Fitting

Implemented in `prepare_fit_base` (`src/ecf_model/fitting.py`).

For each fund-quarter the following are built.

8.1 Event indicators

- draw event if drawdown > 0
- repayment event if repayment > 0 and prior NAV above gate
- recallable event if recallable flow > 0 and repayment > 0

8.2 Ratio metrics

- draw ratio = draw amount / max commitment
- repayment ratio = repayment amount / prior NAV (only with NAV gate)

- recallable ratio conditional on repayment

8.3 NAV residual return (omega)

Raw omega is reconstructed from NAV accounting identity:

$$\omega_{obs} = NAV_t / (NAV_{t-1} + Draw_t - Repay_t) - 1$$

Then clipped to configured fitting bounds.

8.4 Buckets

Rows are tagged by:

- strategy
- grade
- age bucket
- size bucket

These buckets drive hierarchical fitting and fallback.

8.5 Calibration bucket map (all calibration layers)

The table below is the exact bucketing used in code for every calibration step.

Calibration step	Bucket(s) used for estimation	Fallback when sparse
Timing probabilities (<code>fit_timing_probs</code>)	global -> strategy -> strategy + age -> strategy + grade + age -> strategy + grade + age + size	Parent level prior + smoothing
Ratio marginals (<code>fit_ratio_models</code>)	strategy + grade + age + size, strategy + grade + age, strategy + age, strategy, global	Use less granular level at lookup; use empirical mean when fit is sparse/unstable
NAV omega regression (<code>fit_omega_models</code>)	strategy + grade + age + size, strategy + grade + age, strategy, global	Use less granular omega level at simulation lookup
Holdout global deltas (<code>_compute_deltas</code>)	All funds pooled (no bucket split)	Neutral defaults
Holdout group deltas (<code>_compute_group_deltas</code>)	strategy + grade	Shrink toward global delta; skip buckets with insufficient holdout obs

Calibration step	Bucket(s) used for estimation	Fallback when sparse
Recallable propensity (<code>_recall_propensity_by_group</code>)	Fund-level recallable rate by <code>strategy + grade</code>	Global propensity
Pre-end repayment multipliers (<code>_compute_pre_end_repayment_multipliers</code>)	<code>strategy + pre_end_phase</code> where phase is far, mid, near	ALL <code>strategy</code> for same phase, else neutral
Endgame repayment ramp (<code>_compute_endgame_ramp</code>)	<code>strategy</code>	ALL <code>strategy</code>
Post-invest draw multipliers (<code>_compute_post_invest_draw_multipliers</code>)	<code>strategy</code>	ALL <code>strategy</code>
Tail floors/caps/drag (<code>_compute_tail_floors</code>)	<code>strategy + tail_bucket</code> where tail bucket is 0-3, 4-7, 8-11, 12+	Strategy aggregate then ALL <code>strategy</code> same tail bucket
Lifecycle phase multipliers + NAV floor (<code>_compute_lifecycle_multipliers</code>)	<code>strategy + grade + life_phase</code> where life phase is far, mid, late, final, post_0_4, post_5_8, post_9_12, post_12p	<code>strategy + ALL grade + phase</code> then ALL + ALL + phase
Neutral tuning targets (<code>_tune_neutral_calibration_strategy</code>)	Primary target: <code>strategy + grade</code> ; secondary/global alignment: pooled portfolio	Global intensity scales after group-level updates

Simulation lookup order is deterministic:

- Timing/ratio: `strategy+grade+age+size -> strategy+grade+age -> strategy+age -> strategy -> global`
- Omega: `strategy+grade+age+size -> strategy+grade+age -> strategy -> global`
- Lifecycle: `strategy+grade+phase -> strategy+ALL+phase -> ALL+ALL+phase`

9. Timing Probability Model

Function: `fit_timing_probs`.

9.1 What it estimates

Per bucket it estimates quarterly probabilities for:

- draw event
- repayment event
- recallable event

9.2 Hierarchy

Levels:

- global
- strategy
- strategy + age
- strategy + grade + age
- strategy + grade + age + size

9.3 Smoothing

Raw rates are smoothed using parent priors and a configured prior strength:

$$p = (events + prior_strength * p_{parent}) / (n + prior_strength)$$

This prevents unstable small buckets from overfitting.

9.4 Diagnostics

Wilson confidence intervals are stored for review.

10. Ratio Marginal Model

Function: `fit_ratio_models`.

10.1 Ratios modeled

- draw size ratio
- repayment size ratio
- recallable ratio conditional on repayment

10.2 Procedure

For each bucket and metric:

1. Keep positive observations.
2. Winsorize tails.
3. Fit lognormal and gamma candidates.
4. Select by AIC.
5. If sparse, use empirical mean fallback.

10.3 Why this approach

- Lognormal and gamma both handle positive skew common in private-market cashflow sizes.
 - AIC selection is simple and transparent.
 - Empirical fallback ensures coverage in sparse regions.
-

11. NAV Dynamics Model (Omega)

Function: `fit_omega_models`.

11.1 Target

`omega_obs` from reconstructed NAV residual behavior.

11.2 Predictors

- intercept
- current MSCI return
- lagged MSCI return
- age and age squared
- time to end of life
- pre-end indicator
- post-end indicator

11.3 Estimation

Robust ridge regression:

- L2 regularization
- robust reweighting (Tukey bisquare)
- hierarchy-specific penalty strengths

11.4 Outputs

For each bucket:

- coefficients
- residual sigma
- in-sample RMSE
- in-sample R2

11.5 Practical interpretation

In current runs, market betas are generally positive at broad levels, while fine-bucket coefficients can be noisy. This is typical under sparse private-market panel structure.

12. MSCI Model

Implemented in `src/ecf_model/msci.py`.

12.1 Fit stage

- Quarterly returns from index levels
- AR persistence estimate
- 3 regimes by return terciles
- Regime-specific mean/volatility
- Regime transition matrix with smoothing

12.2 Projection stage

Each simulation path evolves by:

1. transition regime
2. draw shock
3. apply AR + regime drift + scenario drift
4. update index level

12.3 Scenario tilts

- bullish: positive drift shift + transition tilt toward bullish regime
 - bearish: negative drift shift + transition tilt toward bearish regime
 - neutral: baseline
-

13. One-Factor Copula Dependence

13.1 Motivation

Without dependence, event timing and ratio draws are too independent within a fund, producing unrealistic smoothness.

13.2 Calibration

Rho is calibrated from historical lag-1 dependence signatures and saved to `copula_calibration.json`.

13.3 Application

For each fund and simulation path:

- draw one common latent factor
- combine with idiosyncratic noise per random component

- map to uniform via normal CDF
- use same mechanism for event draws and ratio inverse CDF draws

13.4 Interpretation

- dependence is within-fund across random components
- cross-fund cashflow dependence is not explicitly modeled via copula
- cross-fund co-movement primarily comes through shared MSCI paths

14. Holdout Recalibration Framework

Primary function: `calibrate_from_history`.

14.1 Goal

Adjust base fit outputs toward recent historical behavior without overfitting.

14.2 Split

- Train: earlier data
- Holdout: recent quarters before cutoff

14.3 Global adjustments estimated

- draw event multiplier
- repayment event multiplier
- draw ratio scale
- repayment ratio scale
- omega bump
- repayment timing shift

14.4 Significance gates

If holdout difference is not statistically significant, raw effect is neutralized.

14.5 Shrinkage

Each effect is shrunk by sample size using $n/(n + shrink_n)$.

14.6 Clipping

Final effects are clipped to configured safe intervals.

15. Stability Attenuation Framework

Purpose: reduce deltas that look unstable over time.

Method:

- Build rolling pseudo-out-of-sample windows.
- Measure how well one period's observed effect predicts next period's effect.
- Fit slope-through-origin in $[0, 1]$ for each metric.
- Multiply global delta magnitude by attenuation slope.

Interpretation:

- slope near 1: stable signal
 - slope near 0: unstable signal, heavily damped
-

16. Group-Level Recalibration (Strategy x Grade)

Function: `_compute_group_deltas`.

16.1 Why group deltas

Portfolio mix is strategy and grade specific. Global corrections are often insufficient.

16.2 How group deltas are formed

For each group:

- compute local holdout deltas
- require minimum holdout observations
- shrink toward global delta
- clip to bounds

16.3 Recallable propensity

Also estimates group-level probability that a fund exhibits recallables at all.

Used in simulation as a per-fund Bernoulli enablement for recallable behavior.

17. Phase-Specific Multipliers: Full Methodology

This section is the most important for lifecycle timing.

17.1 Why phase multipliers exist

Base timing and ratio models capture broad age-driven behavior. They are not sufficient to reproduce:

- pre-end repayment timing shifts
- post-investment drawdown decay
- post-end runoff profiles
- residual NAV persistence

Phase multipliers are calibrated overlays learned from history and applied systematically.

17.2 Common estimation recipe

For each phase table:

1. define phase windows
2. compute phase behavior and baseline behavior
3. compute raw phase effect
4. run significance tests
5. neutralize if not significant
6. shrink by sample size
7. clip to configured ranges
8. build fallback rows (strategy/global)

17.3 Pre-end repayment multipliers

File: `pre_end_repayment_multipliers.csv`.

Phases:

- far ($q_t o_e nd > 24$)
- mid ($13 \leq q_t o_e nd \leq 24$)
- near ($1 \leq q_t o_e nd \leq 12$)

Behavior:

- far and mid can attenuate repayment frequency and size
- near is forced neutral here to avoid overlap with endgame ramp

Applied when far/mid pre-end only.

17.4 Endgame repayment ramp

File: `endgame_repayment_ramp.csv`.

Compares near-end window (1..12) with mid pre-end (13..24) and derives multipliers.

Simulation uses a linear ramp so effect increases smoothly as `q_to_end` approaches 1.

17.5 Post-invest draw multipliers

File: `post_invest_draw_multipliers.csv`.

Compares draw behavior in:

- in-investment period
- after investment period, before end

By design this layer is attenuation only (multipliers clipped at or below 1 for post-invest behavior).

17.6 Tail floors (post-end)

File: `tail_repayment_floors.csv`.

Post-end buckets:

- 0-3
- 4-7
- 8-11
- 12+

Per bucket, strategy-level controls:

- minimum repayment probability
- minimum repayment ratio
- maximum draw probability
- omega drag

These controls are critical to avoid both unrealistic immediate liquidation and unrealistic late-life draw behavior.

17.7 Lifecycle phase multipliers

File: `lifecycle_phase_multipliers.csv`.

Phases:

- far
- mid
- late
- final
- post_0_4
- post_5_8
- post_9_12
- post_12p

For each strategy-grade-phase, table contains:

- repayment probability multiplier
- repayment ratio multiplier
- draw probability multiplier

- draw ratio multiplier
- omega bump
- NAV floor ratio

17.8 NAV floor ratio estimation

The model estimates a historical NAV-to-commit quantile in late/post phases and uses it as floor anchor.

If data is weak in a phase, fallback anchors are used from broader levels.

17.9 Why this is defensible

These phase layers are not arbitrary constants. They are:

- estimated from observed phase behavior
 - significance-gated
 - shrunk by sample size
 - bounded by explicit stability clips
-

18. Simulation Engine: Quarter-By-Quarter Mechanics

The simulator executes nested loops over simulation path, quarter, and fund.

18.1 Initial state

Per fund at projection start:

- commitment
- NAV
- cumulative draw/repay/recallable
- age
- lifecycle dates

Recallable redraw balance is initialized from cumulative history.

18.2 For each quarter and fund

1. Determine lifecycle flags:
 - pre-first-close
 - in-investment
 - after-invest pre-end
 - in-endgame
 - post-end
2. Pull base timing probabilities by hierarchical fallback.
3. Apply global/group recalibration multipliers.
4. Apply lifecycle multipliers (strength-weighted).

5. Apply pre-end attenuation for far/mid.
6. Apply endgame ramp for near-end.
7. Apply post-invest draw attenuation.
8. Apply post-end tail floors/caps/drag.
9. Draw events and sample amounts.
10. Update cashflow cumulatives and recallable balance.
11. Compute omega and update NAV.
12. Apply residual NAV floor pull if applicable.

18.3 Drawdowns

- require draw allowed state
- event sampled from adjusted draw probability
- amount sampled from draw ratio marginal and capped by capacity

18.4 Repayments

- require NAV gate
- event sampled from adjusted repayment probability
- amount sampled from repayment ratio marginal with phase floors and multipliers

18.5 Recallables

- only if repayment occurred
- event sampled from conditional recallable probability
- ratio sampled from recallable conditional marginal
- cumulative recallables capped by cumulative repayments

18.6 NAV update details

NAV after cashflow is multiplied by $(1 + \textit{omega})$ where omega includes:

- fitted coefficients
- MSCI contemporaneous and lagged effects
- lifecycle indicators
- global/group omega bumps
- lifecycle omega bump
- residual random shock
- post-end omega drag

Then clipped and optionally pulled toward lifecycle NAV floor target.

18.7 Aggregated outputs

Simulator saves means and quantiles (portfolio-level) and means (fund-level).

19. Scenario Handling

Scenarios act through MSCI dynamics, not direct cashflow probability table swaps.

Effects:

- bullish: higher drift + bullish transition tilt
- bearish: lower drift + bearish transition tilt
- neutral: base

Because omega uses MSCI returns, scenario effects propagate naturally into NAV and indirectly into repayment amounts.

20. Neutral Tuning Pipeline

Tuning exists to align neutral projections with historical behavior for the specific strategy-grade mix of the target portfolio.

20.1 Target building

Targets may use:

- pooled historical IRR/TVPI (recommended)
- mix-median mode

Additional targets include:

- recallable-to-commit ratio
- draw-to-commit ratio for mature funds

20.2 Objective function

IRR is weighted most heavily, then TVPI and shape error.

20.3 Iterative updates

Group deltas are updated iteratively based on mismatch by strategy-grade.

Adjustments involve:

- repayment ratio scale
- repayment probability
- omega bump
- optional timing shift

20.4 Post-iteration alignment

Coarse/fine scaling searches are run for:

- repayment intensity (IRR alignment)
- recallable intensity
- draw intensity
- final repayment nudge

20.5 Persistence

Tuned calibration artifacts and diagnostics are written to calibration directory used for final projection.

21. Manual Timing Overlay (Timing Profile V2)

Quarter runner applies an optional post-tuning overlay specific to PE/VC timing.

Current parameters include:

- repayment scale adjustment
- far phase relaxation
- endgame relaxation
- tail floor scaling
- late phase scaling

This layer is deterministic and business-driven, applied after statistical tuning.

Implication:

- final behavior is calibrated + tuned + overlaid
 - for pure statistical runs, disable this overlay
-

22. Output Artifacts And Interpretation

22.1 Calibration folder

Key files:

- timing probabilities
- ratio fits
- omega fits
- global/group recalibration deltas
- phase multipliers
- tail controls
- MSCI model
- copula calibration
- tuning diagnostics

22.2 Projection folder

Key files:

- portfolio observation summary
- portfolio simulation series
- fund quarterly means
- fund terminal NAV means
- MSCI path table
- projection config snapshot

22.3 What is and is not persisted

Persisted:

- aggregated cashflow outcomes
- full MSCI paths

Not persisted by default:

- full fund-level full-path cashflow tensors
-

23. Diagnostics And Model Review Checklist

23.1 Pre-run checks

- cutoff quarter consistency across history and MSCI
- portfolio snapshot quarter sanity
- canonical schema validity

23.2 Fit diagnostics

- bucket coverage counts
- distribution selection reasonableness
- omega beta signs and sigma ranges

23.3 Calibration diagnostics

- significance and shrinkage balance
- stability attenuation values
- phase table coverage and fallback usage

23.4 Projection diagnostics

- cumulative drawdowns vs commitment expectations
- repayment timing shape
- residual NAV tail behavior
- DPI/TVPI/IRR trajectory plausibility

23.5 Portfolio-level comparatives

When possible, compare against historical comparable portfolio slices by strategy, grade, vintage, and maturity filters.

24. Statistical Defensibility: Learned Vs Imposed

24.1 Learned from historical data

- timing probabilities
- ratio marginals
- omega coefficients
- holdout deltas
- phase multipliers
- tail controls
- copula rho
- overrun estimates

24.2 Controlled policy knobs

- clipping ranges
- shrinkage strength
- prior strength
- minimum observation thresholds
- scenario drift shifts
- optional manual overlay

24.3 Defensibility statement

The model is defensible as a constrained statistical system with transparent business controls. It is not purely data-driven end-to-end.

25. Risk, Limitations, And Failure Modes

25.1 Sparse bucket instability

Very granular buckets can produce noisy coefficients. Hierarchical fallback mitigates but does not fully remove this risk.

25.2 End-date misspecification

If fund end dates are biased, phase logic may activate at wrong times.

25.3 Scenario concentration

Scenario effects enter mainly through MSCI and omega. If cashflow timing itself is highly scenario-sensitive beyond NAV pathways, this may understate/overstate effects.

25.4 Recallable data quality

Misinterpretation of cumulative vs flow recallables can bias both fit and simulation.

25.5 Manual overlay governance

If overlays are changed frequently without controlled review, reproducibility and defensibility weaken.

26. Operational Runbooks

26.1 One-command quarter run

- Use `scripts/run_quarter_pipeline.py` or OS wrappers.
- Inputs:
 - historical dataset
 - MSCI file
 - portfolio(s)
 - cutoff

26.2 Projection-only run

- Use `python -m ecf_model.cli project`
- Provide portfolio input and calibration directory.

26.3 Reusing calibration

When reusing calibration across test portfolios:

- keep same cutoff basis
 - verify strategy/grade mix comparability
 - preserve scenario and seed settings for reproducible comparisons
-

27. Troubleshooting Guide

27.1 No module named `ecf_model`

Environment not installed with editable package or wrong interpreter active.

27.2 Zero or missing outputs early quarters

Often caused by projection start date, first-close gating, or empty active fund state after filters.

27.3 Drawdowns too low

Check:

- draw multipliers
- post-invest draw attenuation
- mature draw alignment in tuning diagnostics

27.4 Repayments too low/high

Check:

- pre-end multipliers
- endgame ramp
- lifecycle repayment multipliers
- tail floors
- manual timing overlay values

27.5 NAV too low/high at end

Check:

- omega coefficients and sigma
- post-end omega drag
- lifecycle NAV floor ratios
- hard-zero behavior thresholds

28. Governance, Change Control, And Documentation Standards

Recommended governance process:

1. classify change type:
 - statistical parameter change
 - structural code change
 - business overlay change
2. run standard diagnostics and backtests
3. capture before/after metrics
4. update methodology docs and run config snapshot
5. archive calibration artifacts for traceability

Minimum reproducibility package per major run:

- commit hash
 - full run config json
 - calibration directory snapshot
 - key projection outputs
-

29. Suggested Enhancements

Priority enhancements:

1. Optional export of full simulation cashflow paths.
 2. Cross-fund latent factor for cashflow dependence.
 3. Bayesian hierarchical shrinkage for omega coefficients.
 4. Expanded distribution families for ratio tails.
 5. Automated calibration governance report.
 6. Scenario-aware cashflow timing overlays beyond MSCI channel.
-

30. Appendix A: Formula Reference

A.1 Timing smoothing

$$p = (events + prior_strength * p_{parent}) / (n + prior_strength)$$

A.2 Draw ratio

$$draw_ratio = draw_abs / commitment_max$$

A.3 Repayment ratio

$$rep_ratio = repay_abs / nav_p_rev \text{ (with NAV gate)}$$

A.4 Recallable conditional ratio

$$rc_ratio = recall_flow / repay_abs$$

A.5 Omega reconstruction

$$omega_abs = NAV_t / (NAV_{t-1} + Draw_t - Repay_t) - 1$$

A.6 Copula transform

$$Z = \sqrt{rho} * Z_{common} + \sqrt{1 - rho} * \epsilon$$

$$U = \Phi(Z)$$

A.7 Shrinkage pattern

$$adjusted = 1 + w * (raw - 1)$$

$$w = n / (n + shrink_n)$$

A.8 Endgame ramp

$$effective = 1 + (base_{mult} - 1) * ramp$$

$$ramp = (12 - q_{end}) / 11$$

A.9 Objective (neutral tuning)

$$obj = 8 * |IRR_{error}| + 0.6 * |TVPI_{error}| + shape_{error}$$

31. Appendix B: Artifact Reference

Common calibration files:

- timing_probs_selected.csv
- ratio_fit_selected.csv
- omega_selected.csv
- holdout_recalibration.json
- holdout_group_recalibration.csv
- stability_attenuation.csv
- pre_end_repayment_multipliers.csv
- endgame_repayment_ramp.csv
- post_invest_draw_multipliers.csv
- tail_repayment_floors.csv
- lifecycle_phase_multipliers.csv
- msci_model.json
- copula_calibration.json
- neutral_tuning_diagnostics.csv
- timing_adjustment_applied.json (if overlay applied)

Common projection files:

- portfolio_observation_summary.csv
 - sim_outputs/sim_portfolio_series.csv
 - sim_outputs/sim_fund_quarterly_mean.csv
 - sim_outputs/sim_fund_end_summary.csv
 - sim_outputs/msci_paths.csv
 - projection_run_config.json
-

32. Appendix C: Worked Quarter Example

This is a conceptual walkthrough for one fund in one quarter.

1. Determine whether fund is before first close, in investment period, near end, or post-end.
2. Pull base event probabilities from timing table for fund strategy/grade/age/size.
3. Apply global and group deltas from holdout recalibration.
4. Apply lifecycle multipliers with configured strengths.
5. If far/mid pre-end, apply pre-end repayment attenuation.
6. If near end, apply endgame ramp.
7. If post-invest pre-end, apply draw attenuation.
8. If post-end, apply tail floors and omega drag.
9. Draw stochastic event uniforms via copula transform.
10. If draw event occurs, sample draw ratio from chosen marginal and compute draw amount with caps.
11. If repayment event occurs, sample repayment ratio and compute repayment amount with floors/caps.
12. If recallable event occurs, sample recall ratio and cap cumulative recallables by cumulative repayments.
13. Compute NAV after flows.
14. Compute omega from regression + shocks + bumps.
15. Update NAV and apply lifecycle NAV floor pull if active.
16. Advance age and cumulative states.

This repeats each quarter across all funds and all simulation paths.

33. Appendix D: Practical Calibration Interpretation

D.1 If far pre-end repayment multipliers are below 1

Interpretation: historically, funds in that stage distribute less frequently or in smaller chunks than broad baseline.

D.2 If endgame ratio multiplier is above 1

Interpretation: as funds approach end, repayment chunks become larger relative to NAV.

D.3 If post-invest draw multipliers are below 1

Interpretation: drawdown activity decays after investment period.

D.4 If tail repayment floor is near zero

Interpretation: very little expected post-end runoff for that strategy-tail bucket.

D.5 If lifecycle NAV floor ratio is high

Interpretation: historical residual NAV tends to remain meaningful late/post end; model preserves this instead of forcing liquidation.

34. Appendix E: FAQ

E.1 Is the model fully data-driven?

Core mechanics are calibrated from historical data, but stability constraints and optional manual overlays are intentionally included.

E.2 Why can TVPI be decent while IRR is low?

Because IRR is highly timing-sensitive; slow repayments and residual NAV concentration can reduce IRR even with acceptable terminal multiples.

E.3 Why do drawdowns sometimes look low?

Usually due to post-invest attenuation, draw alignment scaling, and lifecycle restrictions around maturity.

E.4 Why is there still NAV after fund end?

Historical data shows residual NAV persistence in many strategy-grade slices. The model intentionally preserves that pattern.

E.5 Can copula rho be user-controlled?

Yes. If not provided, it auto-loads calibrated value from calibration artifacts.

E.6 Do we save every simulated cashflow path?

Not by default for fund cashflows. Portfolio aggregates and full MSCI paths are saved.

Closing

This document is intended to be both a technical reference and an audit-ready model narrative.

If you need an even more formal package for governance committee review, the next step is to add:

- fixed benchmark backtest report templates
- parameter drift report between calibration versions

- standardized exception thresholds for override approval

35. Extended Mathematical Detail

This section rewrites the major model blocks with explicit mathematics and implementation notes.

35.1 Notation

Let:

- i index funds
- t index quarters
- s index simulation paths
- C_i commitment
- $N_{i,t}$ NAV at quarter end
- $D_{i,t}$ drawdown flow in quarter
- $R_{i,t}$ repayment flow in quarter
- $K_{i,t}$ recallable flow in quarter
- $A_{i,t}$ age in quarters
- $QTE_{i,t}$ quarters to end of life

Derived indicators:

- $I_{draw,i,t}$ draw event
- $I_{rep,i,t}$ repayment event
- $I_{rc,i,t}$ recallable event (conditional framework)

35.2 Timing event generation

The simulator starts from base probabilities from hierarchical timing tables:

- $p_{draw_base}(i,t)$
- $p_{rep_base}(i,t)$
- $p_{rc_base}(i,t)$

Then applies multiplicative adjustments from multiple layers:

$$p_{draw} = clip(p_{draw_base} * delta_{draw_group} * life_{draw} * post_{invest_draw} * tail_{draw_caplogic}, 0, 1)$$

$$p_{rep} = clip(p_{rep_base} * delta_{rep_group} * life_{rep} * pre_{end_rep} * endgame_{rep} * tail_{rep_floorlogic}, 0, 1)$$

$$p_{rc_cond} = transform(p_{rc_base}, p_{rep_base}, rc_{propensity}, delta_{rc_group})$$

Implementation detail:

- recallable probability is represented as a conditional probability on repayment by converting a joint signal from timing table.

35.3 Ratio generation

For positive-event amount models, ratios are generated from bucket-specific marginals:

- draw ratio marginal
- repayment ratio marginal
- recallable ratio given repayment marginal

The sampled ratio is then scaled by multipliers from recalibration and phase layers.

For each ratio type:

$$ratio_{final} = ratio_{sampled} * delta_{ratio_group} * phase_{ratio_multipliers}$$

and clipped where needed via downstream caps.

35.4 Drawdown amount equation

Given draw event and available drawable capacity:

$$cap_{draw_{i,t}} = \max(C_i + recall_{balance_{i,t}} - cumulative_{draw_{i,t}}, 0)$$

$$denom_{draw_{i,t}} = \max(C_i + recall_{balance_{i,t}}, 0)$$

$$D_{i,t} = \min(cap_{draw_{i,t}}, ratio_{draw_{i,t}} * denom_{draw_{i,t}}, hard_{cap_expression})$$

where hard cap expression uses configured draw ratio cap logic.

35.5 Repayment amount equation

Given repayment event:

$$R_{i,t} = \min(ratio_{rep_{i,t}} * N_{i,t-1}, N_{i,t-1} + D_{i,t})$$

Additional floors in post-end phases:

$$ratio_{rep_{i,t}} = \max(ratio_{rep_{i,t}}, tail_{rep_{ratio_floor}})$$

35.6 Recallable amount equation

Given repayment amount and recallable event:

$$K_{i,t} = R_{i,t} * ratio_{recall_{i,t}}$$

Conservation rule:

$$cumK_{i,t} \leq cumR_{i,t}$$

Hence:

$$K_{i,t} = \min(K_{i,t}, (cumR_{i,t-1} + R_{i,t}) - cumK_{i,t-1})$$

35.7 NAV transition equation

Pre-omega NAV:

$$N_{pre_{i,t}} = \max(N_{i,t-1} + D_{i,t} - R_{i,t}, 0)$$

Omega model:

$$\omega_{i,t} = a + b_0 * MSCI_t + b_1 * MSCI_{t-1} + b_{age} * age + b_{age2} * age^2 + b_{qte} * qte + b_{pre} * I_{pre} + b_{post} * I_{post} + \delta_{omega} + \delta_{life_{omega}} + \epsilon$$

$$\epsilon \sim N(0, \sigma_{bucket})$$

Post-tail drag and clipping:

$$\omega_{i,t} = \text{clip}(\omega_{i,t} + \text{tail}_{omega_drag_post_end}, \omega_{min}, \omega_{max})$$

NAV update:

$$N_{raw_{i,t}} = \max(N_{pre_{i,t}} * (1 + \omega_{i,t}), 0)$$

Then optional post-end tiny NAV hard-zero and optional lifecycle NAV floor pull.

35.8 Lifecycle NAV floor pull equation

For eligible late/post phases:

$$target_{floor_{i,t}} = nav_{floor_ratio_phase} * C_i$$

If $target_{floor} > N_{raw}$, apply partial pull:

$$N_{i,t} = N_{raw_{i,t}} + \lambda_{phase} * (target_{floor_{i,t}} - N_{raw_{i,t}})$$

Else:

$$N_{i,t} = N_{raw_{i,t}}$$

This preserves residual NAV shape while still allowing decay.

36. Detailed Statistical Testing Logic

36.1 Event-rate significance testing

For event frequencies (e.g., repayment event rate), the model uses two-proportion z-tests:

- compare train/baseline event fraction vs holdout/phase event fraction
- if p-value exceeds significance threshold, effect is neutralized

Why this is used:

- event flags are Bernoulli variables
- test is simple and interpretable for governance

36.2 Ratio-level significance testing

For ratio differences, the model uses Welch t-test on transformed values (\log_{1p} in many cases).

Rationale:

- ratio distributions are skewed
- log transform stabilizes variance
- Welch variant tolerates unequal variances

36.3 Neutralization behavior

If statistical support is weak:

- probability multipliers go to 1.0
- ratio multipliers go to 1.0
- omega bumps go to 0.0

This is a key anti-overfit safeguard.

36.4 Shrinkage after significance

Even significant effects are shrunk by sample size:

$$mult_{final} = 1 + w * (mult_{raw} - 1)$$

$$w = n / (n + shrink_n)$$

Meaning:

- small sample -> effect close to neutral
- large sample -> effect closer to raw estimate

36.5 Clipping after shrinkage

Effects are then clipped to configured bounds so one noisy period cannot cause excessive behavior shifts.

37. Bucketing And Hierarchy Design

37.1 Why hierarchy is necessary

Private-market panel data is sparse when segmented by strategy, grade, age, and size simultaneously.

Hierarchy allows:

- detailed behavior where data supports it
- robust fallback when data is thin

37.2 Timing and ratio fallback order

For both timing and ratio lookup, the simulator searches from most specific to broadest:

1. strategy + grade + age + size
2. strategy + grade + age
3. strategy + age
4. strategy only
5. global

37.3 Omega fallback order

Omega lookup similarly falls back from granular to global buckets.

37.4 Implications

- Top-level behavior remains stable even when granular data is sparse.
 - You can safely add funds in underrepresented groups without catastrophic parameter instability.
-

38. Phase-Specific Multipliers: Calculation Walkthrough

This section provides a concrete computation pattern for each phase family.

38.1 Pre-end repayment multipliers

For each strategy and pre-end phase:

1. Build baseline set: all pre-end repayment-gated observations for that strategy.
2. Build phase subset: rows where phase condition holds.
3. Compute:
 - baseline repayment frequency
 - phase repayment frequency
 - baseline conditional repayment ratio
 - phase conditional repayment ratio
4. Raw multipliers:
 - probability multiplier = phase frequency / baseline frequency
 - ratio multiplier = phase ratio / baseline ratio
5. Significance tests for both signals.
6. If non-significant, raw multipliers set to 1.0.
7. Shrink and clip.
8. Near phase forced neutral to avoid overlap with endgame layer.

38.2 Endgame ramp multipliers

For each strategy:

1. define near-end set (1..12 to end)
2. define mid pre-end set (13..24 to end)
3. compute near/mid repayment frequency ratio and repayment size ratio
4. significance gate and positivity check
5. shrink and clip
6. output multipliers

Simulation then applies gradual interpolation from no effect (12 quarters to end) to full effect (1 quarter to end).

38.3 Post-invest draw multipliers

For each strategy:

1. in-invest period sample
2. post-invest pre-end sample
3. compare draw frequency and size
4. significance gate
5. shrink and clip (upper bound prevents increasing post-invest draw intensity)

38.4 Tail floors

For each strategy-tail bucket:

1. collect post-end observations in bucket
2. compute repayment event mean
3. compute repayment ratio from positive repayment rows
4. compute draw event mean
5. compute omega trimmed mean
6. save as floor/cap/drag controls

Then add global fallback rows and fill missing combinations.

38.5 Lifecycle multipliers

For each strategy-grade and lifecycle phase:

1. compute strategy-grade baseline metrics
2. compute phase metrics
3. raw effects = phase vs baseline
4. significance gate each effect separately
5. if phase sample below minimum threshold:
 - neutral behavior for most multipliers
 - nav floor may fallback to anchored level
6. otherwise apply shrinkage and clipping

7. create strategy-level and global fallback rows
 8. complete full phase grid by fallback filling
-

39. Parameter-Level Reference And Practical Impact

This section explains the practical meaning of key configuration parameters.

39.1 Fit parameters

- `min_obs_group`: minimum observations for bucket-level omega fit
- `prior_strength`: smoothing weight in timing probability model
- `ratio_winsor_lower_q`, `ratio_winsor_upper_q`: outlier clipping in ratio fits
- `nav_gate`: minimum prior NAV to allow repayment ratio observations
- `omega_fit_clip`: bounds on omega observations during fit

Practical effects:

- larger prior strength = smoother timing probabilities
- tighter winsor = less tail sensitivity
- higher nav gate = fewer tiny NAV denominator distortions

39.2 Calibration parameters

- holdout length
- significance alpha
- shrinkage strength
- multiplier clip ranges
- lifecycle and phase minimum observations

Practical effects:

- stronger shrinkage = more conservative recalibration
- lower alpha = fewer non-neutral adjustments
- tighter clip ranges = higher stability, lower responsiveness

39.3 Simulation parameters

- number of simulations
- seed
- omega clipping
- copula controls
- phase strengths
- post-end draw policy

Practical effects:

- more sims = smoother estimates, longer runtime

- stronger lifecycle strengths = more pronounced phase behavior
- tighter omega clip = less volatile NAV projection

39.4 Scenario parameters

- drift shift in neutral/bullish/bearish
- volatility scale

Practical effects:

- drift shift mainly affects NAV pathway and therefore repayment capacity over time

40. Extensive Worked Examples

40.1 Example A: Fund in far pre-end phase

Assume a VC fund with:

- age bucket 20+
- far pre-end phase
- post-invest period active

Sequence:

1. Base timing says moderate draw probability and rising repayment probability for age bucket.
2. Far pre-end repayment multiplier < 1 attenuates repayment frequency/size.
3. Post-invest draw multiplier < 1 attenuates draw behavior.
4. Lifecycle far phase may increase draw slightly or dampen repayment depending on group.
5. Result: slower repayment accumulation, moderate NAV persistence.

40.2 Example B: Fund in endgame phase

Assume PE fund 4 quarters to end.

Sequence:

1. Base repayment probability from timing table.
2. Endgame ramp partially active because q_to_end in 1..12.
3. Repayment size multiplier increases as end approaches.
4. Lifecycle late/final multipliers further shape repayment and draw behavior.
5. NAV floor can still keep residual value depending on calibrated ratio.

Result: larger payout chunks but not forced full liquidation unless data supports quick runoff.

40.3 Example C: Post-end runoff

Assume fund 6 quarters after modeled end.

Sequence:

1. Tail bucket 4-7 selected.
2. Repayment probability floored by tail table.
3. Draw probability capped by tail table.
4. Omega drag applied.
5. Hard-zero possible for tiny NAV under threshold.

Result: controlled runoff with decaying NAV and occasional repayments.

41. Full Artifact Column Dictionary

41.1 `timing_probs_selected.csv`

Key columns:

- level
- strategy
- grade
- age bucket
- size bucket
- n_obs
- events_draw, events_rep, events_rc
- p_draw_raw, p_rep_raw, p_rc_raw
- p_draw, p_rep, p_rc
- confidence interval columns

41.2 `ratio_fit_selected.csv`

Key columns:

- metric
- level
- strategy/grade/age/size keys
- n_pos
- mean/p50/p90
- dist (lognorm, gamma, empirical)
- params
- AIC
- KS p-value

41.3 `omega_selected.csv`

Key columns:

- level and bucket keys
- n_obs
- coefficients (a_intercept, b0, b1, age/time/end indicators)
- alpha
- sigma
- in-sample RMSE and R2

41.4 holdout_recalibration.json

Key fields:

- global delta multipliers/scales
- omega bump
- timing shift
- raw diagnostic values
- weights
- p-values
- counts

41.5 holdout_group_recalibration.csv

Per strategy-grade adjustments and weights plus recallable propensity.

41.6 stability_attenuation.csv

Per metric attenuation coefficient and supporting diagnostics.

41.7 pre_end_repayment_multipliers.csv

Per strategy and pre-end phase:

- observation counts
- repayment probability multiplier
- repayment ratio multiplier

41.8 endgame_repayment_ramp.csv

Per strategy:

- near and mid sample sizes
- endgame repayment probability multiplier
- endgame repayment ratio multiplier

41.9 post_invest_draw_multipliers.csv

Per strategy:

- pre and post sample sizes
- draw probability multiplier
- draw ratio multiplier

41.10 tail_repayment_floors.csv

Per strategy-tail bucket:

- repayment floor probability
- repayment floor ratio
- draw probability cap
- omega drag

41.11 lifecycle_phase_multipliers.csv

Per strategy-grade-phase:

- observation counts
- repayment multipliers
- draw multipliers
- omega bump
- nav floor ratio
- p-values for component effects

41.12 sim_portfolio_series.csv

By projection quarter:

- mean draw/repay/nav/recall
- p10/p50/p90 for draw/repay/nav/recall
- p05/p50/p95 for DPI

41.13 portfolio_observation_summary.csv

By quarter:

- quarterly means
- cumulative means
- DPI/RVPI/TVPI to date
- NAV share of total value
- to-date IRR estimate

42. Backtesting And Validation Protocol

42.1 Built-in backtest utility

`src/ecf_model/backtest.py` supports walk-forward backtesting across cutoffs.

42.2 Recommended validation setup

1. choose multiple historical cutoffs
2. fit at each cutoff

3. project active funds to common evaluation quarter
4. compare projected vs realized quarterly draw/repay/nav
5. compute:
 - bias percentages
 - WAPE metrics
 - projected vs actual IRR and TVPI

42.3 What to monitor over time

- directional bias consistency
- calibration drift by strategy
- degradation in tail repayment behavior
- gap between projected and realized NAV persistence

42.4 Governance thresholds (example)

- draw WAPE above threshold triggers review
- sustained repayment bias sign across 3 cutoffs triggers retuning
- omega R2 collapse at broad levels triggers model diagnostics

43. Sensitivity Analysis Playbook

43.1 Why sensitivity matters

Point calibration can look good while being fragile to small assumptions.

43.2 Recommended stress axes

- copula rho
- omega clip bounds
- lifecycle strengths
- endgame multipliers
- tail floors
- scenario drift shifts

43.3 Example sensitivity grid

Run neutral with:

- rho in [auto, 0.1, 0.2, 0.35, 0.5]
- lifecycle NAV floor strengths at low/base/high
- endgame ramp suppressed/base/boosted

Compare:

- cumulative draw/repay
- end NAV

- DPI/TVPI/IRR
- phase timing shape metrics

43.4 Interpretation

If key outcomes move too much under minor knob changes, calibration is under-identified and requires stronger shrinkage or simpler structure.

44. Model Risk Inventory

44.1 Data risks

- missing/incorrect dates
- grade misclassification
- recallable interpretation mismatch
- NAV quality breaks

44.2 Statistical risks

- sparse bucket instability
- significance false positives in repeated testing
- overfitted phase effects if shrinkage too weak

44.3 Structural risks

- end-date estimation errors
- too strong manual overlay
- scenario assumptions not matching regime shifts

44.4 Operational risks

- running with stale calibration against updated data
- accidental overwrites from fixed run tags
- environment mismatches across machines

44.5 Mitigations

- standardized run config snapshots
 - artifact versioning
 - backtest suite before production handoff
 - documented override governance
-

45. Governance Template For Parameter Changes

When proposing a parameter change:

1. describe objective and hypothesis
2. list exact parameters changed
3. run baseline and changed model on same seed/sims
4. compare portfolio and strategy-grade impacts
5. provide backtest comparison
6. record decision and rationale

Suggested decision table columns:

- parameter name
 - old value
 - new value
 - expected direction of impact
 - observed impact on draw/repay/nav/IRR
 - observed impact on backtest error
 - approval status
-

46. Release Checklist

Before declaring a model release:

- data cutoff confirmed and consistent
 - fit artifacts generated cleanly
 - tuning diagnostics reviewed
 - scenario outputs reviewed
 - Sentral and primary test portfolio runs validated
 - backtest suite executed and compared to previous release
 - documentation updated
 - run reproducibility confirmed from clean environment
-

47. Practical Interpretation Guide For Stakeholders

47.1 For investment teams

Focus on:

- repayment timing by lifecycle stage
- residual NAV realism near fund end
- strategy/grade differentiated outcomes

47.2 For finance teams

Focus on:

- cumulative draw and repay envelopes
- liquidity timing windows
- downside and upside scenario bands

47.3 For risk/governance teams

Focus on:

- learned vs imposed parameter split
- stability attenuation values
- backtest errors and trend

47.4 For engineering teams

Focus on:

- deterministic run configuration
 - artifact integrity checks
 - environment parity across OS
-

48. Extended FAQ

48.1 Why not fit one giant black-box model?

Private-market data sparsity and lifecycle heterogeneity often make giant models opaque and unstable. The current architecture keeps components auditable and controllable.

48.2 Why do we need both lifecycle multipliers and endgame ramp?

They capture different scales:

- lifecycle multipliers capture broad phase tendencies
- endgame ramp captures explicit near-end acceleration pattern with smooth timing

48.3 Why cap multipliers?

Caps reduce overreaction to noise and make quarterly re-calibration safer in production.

48.4 Why keep manual timing overlay at all?

It allows policy-level adjustments where historical-only calibration may undershoot operational expectations. It should remain governed and documented.

48.5 Does copula create cross-fund contagion?

Not directly in current design. It creates within-fund dependence. Cross-fund co-movement mainly comes from common MSCI scenario paths.

48.6 Why can post-end repayments still be material?

Because history often shows extended runoff. Tail floors encode this behavior by strategy and elapsed post-end time.

48.7 Why might drawdowns stay below commitment?

Because mature funds historically may not call 100 percent, and post-invest attenuation plus draw caps can reduce final called ratio.

48.8 Is pooled IRR always the right target?

It is strong for portfolio-level calibration, but should be complemented by shape checks, strategy-grade consistency, and backtest validation.

49. Implementation Map By Function

49.1 Core pipeline

- `run_fit_pipeline`: fit and calibration artifact generation
- `run_projection_pipeline`: projection simulation and output persistence

49.2 Fit helpers

- `prepare_fit_base`
- `fit_timing_probs`
- `fit_ratio_models`
- `fit_omega_models`

49.3 Calibration helpers

- `_compute_deltas`
- `_compute_group_deltas`
- `_compute_stability_attenuation`
- `_compute_pre_end_repayment_multipliers`
- `_compute_endgame_ramp`
- `_compute_post_invest_draw_multipliers`
- `_compute_tail_floors`
- `_compute_lifecycle_multipliers`

49.4 Simulation helpers

- `_Lookup` hierarchical retrieval logic
- `_copula_uniform` latent-to-uniform transformation
- `_sample_ratio` inverse-CDF draws for lognormal/gamma
- lifecycle and tail application logic in simulation loop

49.5 Tuning helpers

- target construction functions
 - objective function
 - scaling and iterative update routines
-

50. Long-Form Narrative: How One Full Run Is Built

This section narrates a full quarter run as a sequence of model construction decisions.

1. **Cutoff is fixed.** All historical learning is frozen at that quarter. This establishes the training universe and protects against look-ahead.
2. **Base relationships are learned.** The model learns event frequencies, size distributions, and NAV sensitivities for many buckets.
3. **Recent-period correction is learned.** Holdout recalibration asks: are recent quarters behaving differently from earlier history? If yes, by how much, and with what confidence?
4. **Unstable corrections are damped.** Stability attenuation tests whether observed corrections persist over rolling windows.
5. **Lifecycle realism is imposed from data.** Specialized phase tables encode where behavior reliably changes over life cycle.
6. **Scenario paths are generated.** Market paths are simulated under selected scenario assumptions.
7. **Fund states are initialized.** Each fund starts from latest observed balances and lifecycle positions.
8. **Quarter simulation proceeds.** Event probabilities and ratio sizes are produced from layered multipliers and copula dependence.
9. **NAV is updated consistently.** NAV accounting identity plus omega process drives next-quarter NAV.
10. **Residual NAV is preserved where supported by data.** Late-life floor logic prevents unrealistic forced liquidation patterns.
11. **Aggregates are produced and saved.** Portfolio-level and fund-level outputs are persisted for diagnostics and reporting.

12. **Optional tuning aligns portfolio to historical targets.** If enabled, strategy-grade and portfolio-level corrections are iteratively refined.
 13. **Optional manual timing profile overlay applies policy preferences.** This final layer can smooth or reshape repayment timing for practical alignment.
 14. **Final projection artifacts are published.** These files are the basis for KPI review and decision-making.
-

51. Extended Practical QA Checklist

Use this checklist before accepting a run.

51.1 Data QA

- Are all required columns present and non-empty where expected?
- Are quarter fields consistent and parseable?
- Are commitment and NAV scales plausible?
- Is recallable interpretation correct (flow vs cumulative)?

51.2 Fit QA

- Are timing probabilities in sensible ranges by age?
- Are chosen ratio distributions plausible and not dominated by sparse empirical fallbacks in key buckets?
- Are omega broad-level betas directionally sensible?

51.3 Calibration QA

- Do global deltas show realistic magnitudes?
- Are group deltas concentrated only where data support exists?
- Are stability attenuations reasonable (not all 0, not all 1)?
- Do phase tables have expected directional behavior?

51.4 Projection QA

- Drawdown trajectory vs commitment expectation
- Repayment buildup and tapering behavior
- End NAV and runoff plausibility
- DPI/TVPI/IRR internal consistency

51.5 Sensitivity QA

- Check whether small rho or phase-strength changes produce extreme metric swings
- If yes, revisit shrinkage and clipping before production

52. Concluding Guidance

This system is strongest when used with disciplined process:

- consistent data refresh cadence
- controlled parameter governance
- routine backtesting and sensitivity checks
- explicit documentation of any overlay changes

The architecture is intentionally practical: it combines statistical learning with guardrails that reflect business reality and production reliability requirements.

For audit and governance, preserve every run with:

- full config
 - full calibration artifact folder
 - key projection output files
 - summary diagnostics
-

53. Module-By-Module Implementation Walkthrough

This section maps each module to implementation responsibilities and operational expectations.

53.1 `src/ecf_model/config.py`

What this module does:

- Defines all major dataclass configurations used by fit, calibration, simulation, and scenarios.

Why this matters:

- This file is effectively the model's contract for defaults.
- Every run config snapshot serializes these settings, so drift in defaults changes model behavior.

Key groups:

- Fit config: smoothing and fit-level controls.
- Calibration config: holdout split, significance, shrinkage, clipping.
- Simulation config: Monte Carlo controls, copula, lifecycle strengths.
- Scenario config: drift shifts and volatility scaling.

Operational recommendation:

- Treat default changes as model changes requiring backtest review.

53.2 `src/ecf_model/schema.py`

What this module does:

- Canonical column mapping and required schema validation.
- Quarter parsing support.
- Core type normalization.

Why this matters:

- Most run failures are schema or type issues, not model math issues.
- This module enforces minimum structural integrity.

Operational recommendation:

- If source data provider changes column names, update this mapping first.

53.3 `src/ecf_model/data_io.py`

What this module does:

- File reading from csv/xlsx.
- Canonicalization pipeline orchestration.
- Cutoff slicing.

Critical behavior:

- csv parser supports semicolon and comma patterns.
- `slice_to_cutoff` is strict and should be trusted as training boundary.

Operational recommendation:

- For reproducibility, always log the exact historical file hash and cutoff used.

53.4 `src/ecf_model/utlils.py`

What this module does:

- Quarter utilities, clipping helpers, winsorization, recallable conversion, size bucket assignment.

Most sensitive utility:

- `recallable_to_flow` due to cumulative-vs-flow ambiguity.

Operational recommendation:

- Validate recallable behavior with a random sample of funds whenever upstream extraction changes.

53.5 `src/ecf_model/features.py`

What this module does:

- Builds `FundState` objects used by simulation.
- Estimates investment end and fund end.
- Adds size and age buckets.

Why this is central:

- Lifecycle dates are anchor points for phase logic.
- Poor lifecycle estimation propagates directly into timing distortions.

Operational recommendation:

- Review overrun summaries by strategy every release.

53.6 `src/ecf_model/fitting.py`

What this module does:

- Creates fit base features.
- Fits timing probabilities.
- Fits ratio marginals.
- Fits omega regressions.

Why robust ridge for omega:

- NAV residual behavior is noisy and outlier-prone.
- Robust weighting limits overreaction to extreme observations.

Operational recommendation:

- Monitor broad-level omega coefficients each release to ensure directional stability.

53.7 `src/ecf_model/calibration.py`

What this module does:

- Performs holdout recalibration and generates all phase-specific tables.

Critical safeguards in this module:

- significance gating
- sample-size shrinkage
- clipping
- stability attenuation
- fallback grid completion

Operational recommendation:

- Never bypass significance and shrinkage safeguards in production runs.

53.8 src/ecf_model/msci.py

What this module does:

- Fits and projects MSCI process with regime transitions.

Why this design:

- Keeps scenario behavior explicit and auditable.
- Avoids over-complex time-series model that is harder to govern.

Operational recommendation:

- Validate regime transition matrix and drift assumptions at least quarterly.

53.9 src/ecf_model/simulator.py

What this module does:

- Full quarter-level Monte Carlo simulation.
- Applies all learned and overlay controls in deterministic order.

Why ordering matters:

- Multipliers compound. Changing order can materially alter outputs.

Operational recommendation:

- Preserve operation sequence unless there is a reviewed model change plan.

53.10 src/ecf_model/pipeline.py

What this module does:

- Fit and project orchestration.
- Copula rho resolution.
- Observation summary generation.

Why this matters:

- This is the principal production path.
- It is where run roots and artifact locations are controlled.

Operational recommendation:

- Store the output dictionary from each run as part of run governance logs.

53.11 src/ecf_model/tuning.py

What this module does:

- Neutral tuning to match historical targets for test portfolio mix.

Why this is a separate module:

- Tuning is optional and can be expensive.

- Keeping it separate prevents hidden modifications in standard projection runs.

Operational recommendation:

- Tune only on defined cadence, not ad hoc every run.

53.12 `src/ecf_model/backtest.py`

What this module does:

- Walk-forward evaluation utilities.

Why this matters:

- Backtesting is the primary evidence for defensibility over time.

Operational recommendation:

- Maintain a rolling baseline of backtest outputs and compare release-over-release.

53.13 `src/ecf_model/cli.py`

What this module does:

- command-line entry points for fit/project/project-all/backtest/tune.

Operational recommendation:

- Prefer CLI paths in production scripts to ensure consistent argument handling.

53.14 `scripts/run_quarter_pipeline.py`

What this module does:

- Cross-platform orchestrator for fit+tune+project workflow.
- Snapshot template conversion.
- Optional timing profile overlay.

Operational recommendation:

- Use this for operational quarter runs.
- Use lower-level CLI commands for controlled experiments.

54. Numerical Worked Example: Phase Multipliers

This section demonstrates numerical calculations with simplified values.

54.1 Pre-end repayment attenuation example

Assume strategy X has:

- baseline pre-end repayment event rate: 0.20
- far phase repayment event rate: 0.14

Raw probability multiplier:

$$raw = 0.14/0.20 = 0.70$$

Assume sample size in far phase is 500 and $shrink_n = 800$.

Shrinkage weight:

$$w = 500/(500 + 800) = 0.3846$$

Adjusted multiplier:

$$m = 1 + 0.3846 * (0.70 - 1) = 0.8846$$

If clip range is $[0.6, 1.0]$, final remains 0.8846.

Interpretation:

- far phase repayment probability is reduced by about 11.5 percent versus base.

54.2 Endgame ratio ramp example

Assume endgame repayment ratio multiplier for strategy Y is 1.20.

At $q_{t_{end}} = 12$:

$$ramp = 0$$

$$effective = 1 + (1.20 - 1) * 0 = 1.00$$

At $q_{t_{end}} = 6$:

$$ramp = (12 - 6)/11 = 0.5455$$

$$effective = 1 + 0.20 * 0.5455 = 1.1091$$

At $q_{t_{end}} = 1$:

$$ramp = (12 - 1)/11 = 1$$

$$effective = 1.20$$

Interpretation:

- repayment size uplift activates gradually and reaches full intensity near end.

54.3 Post-invest draw attenuation example

Suppose:

- in-invest draw event rate: 0.45
- post-invest pre-end draw event rate: 0.20

Raw multiplier:

$$0.20/0.45 = 0.4444$$

If significance passes and shrink weight is 0.5:

$$adjusted = 1 + 0.5 * (0.4444 - 1) = 0.7222$$

If clip lower bound is 0.05 and upper 1.0, final is 0.7222.

Interpretation:

- post-invest draw event probability is reduced by ~28 percent from baseline.

54.4 Tail floor example

Post-end bucket 4-7 for strategy Z yields:

- empirical repayment event mean 0.06
- empirical repayment ratio conditional mean 0.04
- draw event mean 0.01
- omega mean -0.03

Simulation effects in that bucket:

- repayment probability cannot fall below 0.06
- repayment ratio cannot fall below 0.04
- draw probability cannot exceed 0.01
- omega gets additional -0.03 drag

Interpretation:

- low but persistent repayment runoff with suppressed draws and NAV decay pressure.

54.5 Lifecycle NAV floor example

Assume lifecycle floor ratio for phase is 0.25 and commitment is 100.

Target floor NAV is 25.

If raw simulated NAV is 15 and floor strength is 0.45:

$$new_{nav} = 15 + 0.45 * (25 - 15) = 19.5$$

Interpretation:

- model does not jump to 25, it moves partially toward floor to preserve dynamics.

55. Detailed Tuning Mechanics With Numeric Intuition

55.1 Objective weighting intuition

The objective places highest weight on IRR mismatch:

$$8 * IRR_error + 0.6 * TVPI_error + shape_error$$

Why:

- business feedback often focuses on timing realism reflected in IRR.
- TVPI is still important but less timing-sensitive.

55.2 Strategy-grade update intuition

If a strategy-grade projected TVPI is below target:

- increase repayment ratio scale slightly
- increase repayment probability mildly
- apply small positive omega nudge if needed

If above target:

- reverse these adjustments.

All updates are clipped to avoid instability.

55.3 Global IRR alignment search intuition

After local updates, global scale search finds repayment intensity level that best closes IRR gap while preserving shape from local calibration.

This decouples:

- local mix shape alignment
- global timing-level correction

55.4 Draw alignment intuition

A separate draw intensity search aligns projected called ratio with mature historical behavior.

This avoids a common failure where IRR improves but called ratio drifts unrealistically.

55.5 Recall alignment intuition

Recallable scaling aligns total recallables to historical share of commitment/repayments for comparable strategy-grade subset.

This preserves consistency of redraw potential.

56. Detailed Runtime And Performance Notes

56.1 Computational complexity

Dominant cost is simulation:

$$O(n_{sims} * n_{quarters} * n_{funds})$$

Cost multipliers:

- copula enabled adds several normal transforms per event/ratio draw
- richer horizon from long lifetimes increases runtime

56.2 Memory profile

Large arrays held:

- per-sim portfolio series matrices
- fund-by-quarter accumulators
- MSCI path matrices

Default settings are chosen to remain practical on laptops while maintaining stable statistics.

56.3 Practical runtime controls

- reduce `n_sims` for exploratory runs
- use fixed seed for A/B comparison runs
- keep tuned final runs at higher sims for stable KPI reporting

56.4 Cross-platform consistency

Given same input files, config, seed, and package versions, outputs should be reproducible within floating-point tolerance across OS.

57. Data Quality Control Procedures

57.1 Pre-fit data checks

Recommended checks before fitting:

- non-null rates for core fields
- duplicate fund-quarter rows
- outlier checks on NAV and flows
- grade and strategy label normalization consistency

57.2 Recallable checks

For random funds:

- inspect raw recallable series shape
- verify monotonic cumulative interpretation where expected

57.3 Lifecycle checks

Review distributions of:

- first close dates
- planned end dates
- estimated overrun by strategy
- resulting fund end dates

57.4 Consistency checks

At fund-quarter level:

- negative commitments should not exist
 - extreme NAV discontinuities should be flagged
 - drawdown and repayment signs should be consistent after canonicalization
-

58. Extended Governance Playbook

58.1 Change categories

Category A: data-only refresh

- no code or parameter change
- expected to preserve structure

Category B: parameter change

- clip bounds, shrinkage, priors, strengths

Category C: structural model change

- equations, sequence, new factors, new distributions

Category D: overlay policy change

- timing profile overlays and scenario drifts

58.2 Minimum evidence by category

A:

- run completion logs
- KPI diff explanation

B:

- A/B run on same seed
- backtest delta summary
- parameter rationale

C:

- full backtest suite
- sensitivity analysis
- review sign-off

D:

- policy rationale
- impact on KPI and timing shape

58.3 Suggested review board artifacts

- one-page executive summary
 - detailed technical appendix
 - reproducibility pack
 - decision log with approvers
-

59. Integration Guide For Portfolio Teams

59.1 Input preparation

If portfolio is not canonical, use quarter runner conversion path.

59.2 Calibration reuse strategy

For multiple test portfolios at same cutoff:

- calibrate once
- reuse calibration directory
- run projection with custom run tags

59.3 Reporting workflow

Recommended portfolio report structure:

- key headline KPIs
- cumulative draw and repay curves
- NAV trajectory
- scenario comparison table
- percentile table for DPI and NAV

59.4 Communication tips

When explaining low IRR with acceptable TVPI:

- emphasize timing concentration and residual NAV weight at horizon

When explaining high end NAV:

- link to lifecycle floor and historical residual NAV patterns
-

60. Internal Consistency Checks

60.1 Cashflow conservation checks

Per fund and portfolio:

- cumulative recallables \leq cumulative repayments
- drawdowns cannot exceed drawable capacity
- repayments cannot exceed available NAV after same-quarter flow

60.2 Monotonicity checks

- cumulative metrics should be non-decreasing
- quarter indices should be contiguous

60.3 Boundary checks

- probabilities in $[0,1]$
- multipliers within configured clips
- omega within simulation clip

60.4 Fallback coverage checks

Ensure each required phase and tail bucket has fallback values to avoid missing-lookup behavior.

61. Long-Form Discussion: Why IRR And TVPI Diverge

IRR and TVPI capture different dimensions.

TVPI reflects total value relative to paid-in capital, regardless of when value arrives.

IRR heavily penalizes delayed distributions and value trapped in residual NAV near horizon.

In this model, divergence can arise when:

- repayments are back-loaded by phase multipliers

- post-end runoff is slow
- lifecycle NAV floors preserve value late

Hence, improving IRR without distorting TVPI often requires timing recalibration rather than terminal multiple recalibration.

62. Long-Form Discussion: Drawdown Ratio Behavior

Drawdown ratio under-mobilization is frequently observed in mature private-market datasets.

Reasons it can appear low in projection:

- historical mature called ratio below full commitment
- post-invest draw attenuation
- end-of-life draw restrictions
- recallable dynamics changing effective drawable balance

The tuning pipeline includes mature draw alignment to mitigate mismatch versus historical mature funds.

63. Extended Scenario Interpretation

63.1 Neutral

Represents statistically central market path assumptions from fitted MSCI process with neutral drift shift.

63.2 Bullish

Adds positive drift and bullish transition tilt.

Typical effects:

- higher omega path
- stronger NAV support
- potentially larger eventual repayments through higher NAV base

63.3 Bearish

Adds negative drift and bearish transition tilt.

Typical effects:

- lower omega path
- weaker NAV support
- slower or smaller repayment profile

63.4 Important caveat

Scenario channel acts mainly through market-sensitive NAV dynamics, not direct replacement of timing probability tables.

64. Detailed Troubleshooting Cases

Case 1: All zeros for early quarters

Likely causes:

- start quarter not aligned with intended snapshot
- many funds pre-first-close at projection start
- no active funds after workflow-stage filtering

Case 2: Unexpectedly high recallables

Check:

- recallable interpretation in source
- group-level recallable multipliers
- recallable scaling from tuning

Case 3: Repayments dropped after enabling copula

Possible reason:

- dependence structure changed event/size co-realization pattern
- compare with same seed and same calibration under copula on/off

Case 4: End NAV too low

Check:

- lifecycle nav floor ratios and strengths
- tail omega drags
- endgame multipliers and pre-end attenuation

Case 5: End NAV too high

Check:

- overly strong lifecycle nav floor
 - weak tail repayment floors
 - low endgame repayment intensity
-

65. Operational Templates

65.1 Model change request template

- request id
- date
- owner
- objective
- exact parameters/files changed
- expected directional impact
- validation evidence
- risk assessment
- approval

65.2 Run log template

- run tag
- cutoff
- scenario
- n sims
- seed
- calibration dir
- key KPI outputs
- notable warnings

65.3 Release note template

- summary of changes
 - rationale
 - KPI deltas vs previous release
 - backtest deltas
 - known caveats
-

66. Extended Appendix: Statistical Caveats

66.1 Multiple testing

Because many buckets and metrics are tested, naive p-value interpretation can overstate confidence. Shrinkage and clipping help mitigate this.

66.2 Non-stationarity

Private-market behavior changes across vintages and macro regimes. Holdout recalibration and stability attenuation are practical but not complete solutions.

66.3 Selection bias

Historical realized profiles can be influenced by reporting and survivorship artifacts.

66.4 Sparse tails

Extreme tail behavior is difficult to estimate robustly in thin buckets. Conservative caps and floors are intentional.

67. Extended Appendix: Potential Future Research Paths

1. Bayesian hierarchical model for timing and ratios with explicit uncertainty intervals.
2. Joint state-space model linking cashflow and NAV evolution under latent quality factors.
3. Cross-fund dependence factor for cashflows beyond market path channel.
4. Dynamic scenario-conditioned timing models rather than static timing + market channel.
5. Formal model confidence sets for parameter uncertainty.

68. Executive Summary For Non-Technical Readers

The model projects private-market cashflows in quarters.

It learns from historical data how funds usually:

- call capital
- return capital
- maintain or reduce NAV

Then it adjusts this behavior by fund lifecycle stage, because late-life and post-end behavior is different from early-life behavior.

It also uses market scenarios through MSCI to influence NAV evolution.

Finally, it runs many simulations and reports average and percentile outcomes.

The model is controlled by safeguards so noisy data does not create unstable projections.

69. Final Governance Recommendation

For production confidence:

- keep manual overlays transparent and versioned

- run walk-forward backtests quarterly
 - maintain a parameter change approval process
 - track calibration drift over time
 - preserve run artifacts for full traceability
-

70. End Note

This document is intentionally exhaustive. It can be used as:

- a technical implementation reference
- a model governance file
- a runbook companion
- a reviewer onboarding manual

For day-to-day use, pair this file with:

- docs/modeling_methodology_deep_dive.md
- docs/modeling_methodology.md