



## Goal & Problem Motivation

### Why This Problem?

- Dynamic mutual-fund networks change with evolving characteristics and fund entry or exit.
- Independent UMAP fits can mix genuine market evolution with arbitrary embedding reorganization.
- Fair temporal comparisons require fund-ID alignment across a changing universe.

### Research Question

Can modular graph construction and stability-constrained optimization improve structural fidelity and temporal consistency in dynamic mutual-fund embeddings?

### Objective

- Separate high-dimensional graph construction from low-dimensional optimization.
- Compare Vanilla, calendar-aligned multi-horizon Memory, and stable-core Bias.
- Evaluate edge persistence, graph geometry, graph fit, and low-dimensional neighborhood stability.

### Key Idea

Hold the optimizer and metrics constant while swapping graph representations. Memory and persistent cores preserve useful structure across consecutive rolling windows.

## Methodology

### Stage 1 - Modular High-Dimensional Graph Construction

**Rolling data:** 263 available monthly tables (Dec. 2000-Dec. 2022), 645 funds, 70 features, and 140-481 funds per timestamp. June and July 2022 are absent. The common evaluation uses 192 available endpoints.

**Vanilla:** current-feature UMAP fuzzy graph, 5 neighbors, Euclidean distance, seed 42.

**Memory:** align current, 12-, 36-, and 60-month feature histories to four 12-by-70 blocks, scale them, concatenate on the feature axis, and flatten to 3,360 coordinates.

*Implementation weights: (w0, w12, w36, w60) = (0.50, 0.30, 0.20, 0.10)*

Missing months and fund-months are zero-padded for shape and marked invalid. A masked Euclidean metric compares only jointly observed entries, so padding cannot create artificial similarity. The representation has no GRU or learned edge state.

**Bias:** build exact-calendar six-month masked raw-cosine graphs, keep the top 10 edges above 0.001, detect strict-intersection cores across 12 centered windows, and reconstruct their union (minimum core size 3).

$$J(E(s), E(t)) = |E(s) \cap E(t)| / |E(s) \cup E(t)|$$

A core chain is retained when its node overlap exceeds the matching threshold across consecutive source windows. Nodes recurring in at least T-n windows become stable anchors for reconstruction.

### Common graph contract

- Return a sparse weighted graph with the corresponding fund IDs.
- Keep graph generation independent from the downstream optimizer.
- Reuse identical alignment, preprocessing, objectives, and diagnostics for every method.

### Modular workflow

- Validate timestamps, feature tables, and unique fund IDs.
- Load or construct Vanilla, Memory, and Bias graphs.
- Optimize each registered loss chronologically.
- Save window-level and aggregate comparisons.

### Registered parameter snapshot

#### Vanilla

Current 70 features  
k = 5; Euclidean

#### Memory

Recency-weighted pooling  
k = 15; 4 horizons

#### Bias

12-window cores  
top-k = 10; union

### Stage 2: Fund-ID Alignment and Graph Preprocessing

Consecutive graphs and embeddings are restricted and reordered by common fund identifiers:

$$V(s,t) = V(s) \cap V(t)$$

- Symmetrize and normalize each graph.
- Prune below the 0.8 edge-weight quantile, renormalize, square weights, and retain top-6 neighbors.
- Initialize a two-dimensional UMAP embedding with 6 neighbors and seed 42.

### Stage 3: Stability-Constrained Embedding Optimization

$$L(t) = \lambda_{\text{task}} L_{\text{task}} + \lambda_{\text{temp}} L_{\text{temp}} + \lambda_{\text{paired}} L_{\text{paired}} + \lambda_{\text{reg}} L_{\text{reg}}$$

Each endpoint runs 200 Adam steps (learning rate 0.05; 5 negative samples). Three registered objectives isolate the effect of temporal and stable-core penalties:

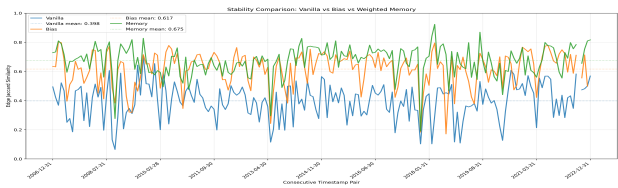
- Task only: (1, 0, 0, 0)
- Temporal 0.01: (1, 0.01, 0, 0)
- Paired 0.01: (1, 0, 0.01, 0)

**Evaluation:** Edge Jaccard; graph-shortest-path HD-LD Pearson correlation; all-pairs normalized cross-entropy; and fund-ID-aligned temporal 15-NN Jaccard. Cross-graph month pairs are excluded.

## Result

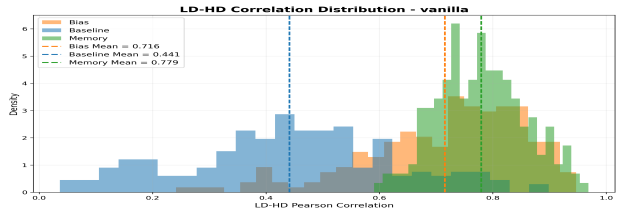
### High-Dimensional Graph Stability

Do graph edges persist across adjacent windows?



Masked Memory has the highest mean Edge Jaccard (0.675), followed by Bias (0.617) and Vanilla (0.398).

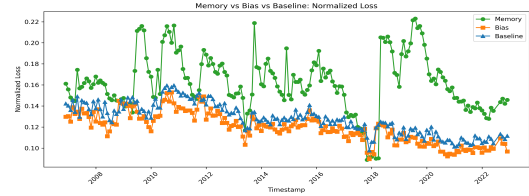
### Graph-Geometry Fidelity



For the task-only objective, Memory has the highest code-generated LD-HD mean (0.779), followed by Bias (0.716) and Baseline (0.441).

### Embedding-to-Graph Diagnostic

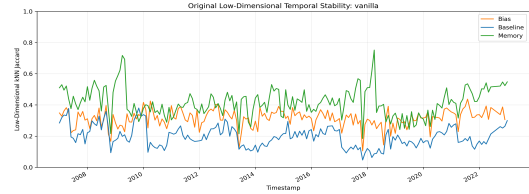
Does each embedding fit its own preprocessed target graph?



Bias has the lowest mean normalized loss (0.119), followed by Baseline (0.128) and Memory (0.162). Targets differ by method, so this remains an internal fit diagnostic rather than a universal ranking.

### Low-Dimensional Temporal Stability

Do aligned 15-nearest-neighbor sets persist between endpoints?



Memory has the strongest mean neighborhood persistence under task-only (0.415), temporal (0.376), and paired (0.376) objectives.

### Interpretation Boundary

Every figure above is embedded directly from the current code output. Edge Jaccard uses the Vanilla fuzzy graph as the blue reference; the other diagnostics use the attention Baseline.

## Registered Evaluation Protocol

- Compare the same 192 available endpoints for all three registered methods.
- Represent edges as fund-ID pairs; align embeddings before any temporal metric.
- Use finite upper-triangle graph distances for HD-LD Pearson correlation.
- Leave missing calendar months as gaps; never compare across a gap.
- Skip temporal kNN pairs with 16 or fewer common funds.

$$\rho(t) = \text{corr}[\text{graph shortest paths}, 2D \text{ Euclidean distances}]$$

Four diagnostics answer distinct questions; no single score is treated as a universal likelihood ranking.

## Overall Comparison

### Memory - strongest stability

Edge Jaccard mean: **0.675**  
Task LD-HD mean: **0.779**  
Task temporal kNN: **0.415**.

### Bias

Edge Jaccard: **0.617**; task LD-HD: **0.716**.  
Lowest normalized loss: **0.119**.

### Vanilla / Baseline

Edge mean: 0.398; task LD-HD: 0.441;  
normalized loss: 0.128.

## Summer 2026 Update

The active Memory encoder uses exact calendar alignment. Missing months and fund-months are zero-padded and masked out of graph distance and downstream initialization.

All comparisons use the masked non-GRU Memory representation. June and July 2022 remain a visible evaluation gap.

## Conclusion & Future Work

The modular framework separates representation quality from optimizer effects. Masked Memory leads temporal stability; Bias leads the method-specific normalized-loss diagnostic and penalized HD-LD means.

### Next steps:

- validate Memory across additional loss settings
- perform 12/36/60-month horizon ablations
- test alternate Bias core-window definitions

### Current limitations:

- one registered Memory horizon mix
- one Bias core configuration
- method-specific loss targets