

Distributed Transform-Domain Representation (DTDR) A Persistent Computational Representation for Machine Learning Systems

Jonathan H. West

1 Introduction

Distributed Transform-Domain Representation (DTDR) is a numerical representation framework in which machine learning parameters and embeddings are stored directly in a globally distributed transform basis.

Unlike conventional floating-point storage, DTDR does not primarily store independently meaningful weights. Instead, it stores a system of globally coupled constraints whose collective solution defines model behaviour.

DTDR is not a compression codec. It is a persistent computational representation. Inference, similarity search, and approximate nearest neighbour (ANN) traversal can operate directly on the stored transform-domain representation without reconstructing full-precision floating-point vectors.

2 Core Architectural Principles

DTDR is built around four structural principles.

2.1 Global Distribution of Information

In DTDR, each stored coefficient contributes to many reconstructed dimensions. No single coefficient corresponds to a semantically isolated parameter.

Consequences include:

- Individual corruption does not cause local catastrophic failure.
- Behaviour depends on global consistency.
- Reconstruction quality degrades smoothly.

2.2 Compute-Capable Storage

Transform-domain representations preserve inner products:

$$q \cdot x = \tilde{q} \cdot \tilde{x}$$

where $\tilde{x} = Hx$ under an orthogonal transform H .

This allows similarity search, ANN routing, scoring, and ranking to operate directly in transform space.

2.3 Graceful Degradation

Because information is globally distributed, corruption produces statistical degradation rather than structural collapse.

Experimental corruption studies show:

- FP16 safetensors fail catastrophically under modest corruption.
- DTDR representations degrade gradually across orders of magnitude greater corruption.

2.4 Residual Compressibility

DTDR transform coefficients exhibit structured regularity.

Empirically:

- INT8 DTDR representations remain 30–35% further compressible under ZIP.
- Conventional FP16 representations show negligible additional compression.

This secondary compressibility is orthogonal to DTDR’s core architecture.

3 Model Storage and Inference

DTDR-compressed models reconstruct to working precision sufficient for inference.

Example (Mistral-7B):

Representation	Size
FP16	~14.5 GB
INT8 DTDR	~6.7 GB

Cosine similarity between FP16 and reconstructed DTDR weights: 0.9998.

Inference throughput remains comparable to FP16 baselines when operating in INT8-compatible kernels.

4 Transform-Domain Similarity Structure

Under an orthogonal transform H :

$$\tilde{x} = Hx$$

Similarity becomes spectrally distributed.

Experiments show:

- Nearest-neighbour identity emerges progressively from dominant spectral coefficients.
- Stability depth correlates with similarity margin.
- L2 certification bounds provide tight early stopping criteria.

Structured embeddings (e.g., GloVe) exhibit stronger spectral concentration than random vectors. This establishes the existence of a genuine trajectory signal in transform space.

5 Hierarchical Trajectory Routing in ANN

The most significant recent result is the integration of transform-domain structure into a full ANN pipeline.

5.1 Within-List Hierarchical Routing

For each IVF list:

- Vectors are grouped into bags of 32.
- A binary tree of segment means is precomputed.
- Node norms are stored.
- Beam descent localises promising leaves.

Each node evaluation costs a single dot product at query time. Leaf candidates are reranked by exact L2 distance.

5.2 Full SIFT1M Benchmark (1M Vectors)

Configuration:

- IVF1024
- Bag size = 32
- Beam width = 2
- 10,000 queries
- Standard recall@10
- Unfiltered ground truth

Results:

$nprobe$	recall@10	Mean Candidates	Flat IVF Candidates
8	0.580	899	~7,812
16	0.657	1,796	~15,625
32	0.698	3,577	~31,250

At $nprobe = 8$:

- Recall matches published IVF1024 flat baseline (~ 0.57).
- Candidate evaluations are reduced by approximately $8-9\times$.

Candidate counts scale predictably as:

$$\text{candidates} \approx nprobe \times 112$$

This deterministic scaling provides operational advantages in latency-sensitive systems.

5.3 Interpretation

Flat IVF evaluates every vector in each probed list. Trajectory routing evaluates only geometrically promising leaf segments.

Increasing the number of selected bags improves recall without increasing candidate counts, confirming that routing is discriminative rather than merely expansive.

The method complements IVF coarse routing and does not attempt to replace graph-based methods such as HNSW.

6 Scope and Limitations

DTDR:

- Is not cryptographic encryption.
- Does not prevent reconstruction if full coefficients are accessible.
- Is not a drop-in replacement for all ANN structures.

It provides a persistent transform-domain computational architecture with hierarchical routing capability and structured candidate reduction.

7 Repository Structure and Reproducibility

Full implementation of trajectory routing experiments:

https://github.com/UnrealJon/DTDR/tree/main/experiments/02_dtdr_end_to_end_search

The repository contains the exact configuration used for the IVF1024 SIFT1M experiments reported here.

8 Intellectual Property

Distributed Transform-Domain Representation (DTDR) and related hierarchical routing methods are the subject of a UK patent application under accelerated examination (Green Channel), UK Patent Application No. GB2602157.6.

The repository is provided for research and evaluation purposes.

9 Conclusion

DTDR is a persistent transform-domain computational representation that:

- Reduces storage
- Preserves inference fidelity
- Exhibits graceful degradation
- Retains secondary compressibility

- Enables hierarchical trajectory routing in ANN pipelines

The full SIFT1M benchmark results demonstrate that transform-domain structure can be leveraged to achieve equivalent recall with substantially fewer candidate evaluations.

DTDR is best understood not as a compression technique, but as an alternative computational architecture for numerical machine learning systems.