DTDR: Distributed Transform-Domain Representation

A Persistent Representation for High-Dimensional Semantic Data

In Brief

DTDR (Distributed Transform-Domain Representation) is a method for persistently storing high-dimensional semantic data in a structured transform-domain form that remains computationally meaningful under quantisation.

Rather than treating transforms as transient execution-time steps, DTDR promotes the transform domain to a first-class, stored-at-rest representation. Information relevant to inference, similarity, and model behaviour is distributed across orthogonal transform coefficients, enabling a range of properties that are difficult or impossible to achieve simultaneously in conventional original-domain formats.

In particular, as a format for persistent high-dimensional semantically encoded data storage, DTDR:

- preserves task-relevant geometric relationships under aggressive quantisation;
- permits reconstruction to a non-identical but functionally sufficient numerical precision compatible with existing inference kernels;
- may support similarity evaluation and other operations directly within the quantised transform domain;
- exhibits robustness and graceful degradation under partial data loss or corruption;
- retains residual structure that enables effective secondary lossless compression;
- admits composite representations formed by sequential structured orthogonal transforms; and
- aligns naturally with memory- and bandwidth-constrained environments, including edge devices that already incorporate generic lossless compression hardware.

DTDR is not a compression codec, encryption scheme, or model-specific optimisation. It is a representation strategy intended to decouple storage precision, transport format, and execution precision, enabling flexible deployment pathways for modern machine-learning systems operating on large volumes of semantic numerical data.

Introduction and Motivation

Modern machine-learning systems increasingly operate on semantically encoded high-dimensional numerical data, including model parameters and vector embeddings. In such representations, meaning, similarity, and inference behaviour arise from geometric relationships between vectors rather than from individual numerical components. As model sizes and embedding collections continue to grow, performance and energy efficiency are increasingly constrained not by arithmetic throughput, but by memory bandwidth, cache utilisation, and data movement between storage and compute.

Conventional approaches to reducing storage and transmission overhead typically rely on compression schemes in which data are encoded into a compact intermediate form that must be fully or substantially reconstructed prior to computational use. Such mandatory reconstruction reintroduces memory traffic and latency, limiting the practical benefits of compression. Reduced-precision quantisation schemes mitigate storage cost, but are generally designed as approximations of conventional representations rather than as first-class stored computational forms, and may require specialised kernels or hardware support to be effective.

DTDR (Distributed Transform-Domain Representation) addresses these limitations by introducing a persistent transform-domain representation intended to serve as the *primary stored form* of high-dimensional semantic data, rather than as a transient intermediate encoding.

What DTDR Is (and Is Not)

DTDR is a method of representing high-dimensional numerical data by applying one or more structured orthogonal transforms, followed by controlled quantisation, and retaining the resulting transform-domain coefficients as the stored-at-rest representation.

Crucially, DTDR is not a compression codec in the conventional sense, nor is it an encryption or obfuscation scheme. While the distributed nature of the transform-domain representation may incidentally provide resistance to casual inspection of stored data files, DTDR does not aim to conceal information or provide security guarantees.

The transform-domain representation is not intended solely for later reconstruction into an original numerical basis. Instead, it is designed to remain computationally meaningful in its stored form, supporting both direct transform-domain computation and functional reconstruction to a working numerical precision when required.

DTDR differs from approaches that apply arbitrary or data-dependent rotations prior to quantisation. The transforms employed are structured, orthogonal, and algorithmically defined (for example, Hadamard or Walsh–Hadamard transforms), admitting efficient evaluation without storage of dense rotation matrices and preserving inner products and geometric structure under projection.

DTDR as a Persistent Representation

In conventional systems, transform-domain representations are typically generated transiently during execution and discarded after reconstruction. DTDR departs from this model by treating the transform-domain representation as the canonical stored form of the data, independent of any specific execution pass.

Once created, a DTDR representation may be stored, transmitted, archived, or reused across multiple computational sessions without reapplication of the transform. This decoupling of storage-time representation, transport-time representation, and execution-time representation enables flexible execution pathways, including:

- reconstruction to a working numerical precision compatible with existing inference kernels and runtimes;
- direct computation on quantised transform-domain coefficients; and
- hybrid approaches combining partial reconstruction with transform-domain operations.

Experimental Support

The properties described above have been validated across a range of experimental evaluations, including semantic similarity search on embedded text corpora, large-scale nearest-neighbour benchmarks, neural network parameter storage, and full large-language model case studies.

These experiments demonstrate preservation of similarity structure under quantisation, graceful degradation under partial coefficient loss, scalability to million-vector datasets and multi-billion-parameter models, and practical reductions in storage footprint and memory bandwidth requirements. Importantly, these effects arise from the representation itself rather than from retraining, learned codebooks, or data-dependent encoding.

Scope and Intended Use

This repository provides a reference implementation and accompanying documentation intended to illustrate the principles and technical effects of DTDR. It is not presented as a production-ready library, but as a foundation for evaluation, discussion, and further development by researchers, engineers, and system architects interested in persistent representations for high-dimensional semantic data.