Ground-Truth-Aware Metric Terminology for Vector Retrieval

Contents

Ground-Truth-Aware Metric Terminology for Vector Retrieval

Author: Sayed Amir Karim (NextX AG)
Organization: NextX AG, Hauptstrasse 20, 6418 Rothenthurm, Switzerland
Contact: compression@aqea.ai
Repository: `aqea-compress`
DOI: 10.5281/zenodo.18152431

How to cite

```
Karim, Sayed Amir (NextX AG).
"Ground-Truth-Aware Metric Terminology for Vector Retrieval."
Version 0.1, 2026-01-05. DOI: 10.5281/zenodo.18152431 (https://doi.org/10.5281/zenodo.18152431).
Reproducibility source: aqea-compress commit 3f73b1a.
```

Primary reference (backlink)

This whitepaper is aligned with AQEA's evidence-backed technical report:

```
Karim, Sayed Amir (NextX AG).
"AQEA: Technical Report - Domain-Adaptive Semantic Compression of Embeddings."
Version 0.1, 2026-01-03. DOI: 10.5281/zenodo.18138436 (https://doi.org/10.5281/zenodo.18138436).
Reproducibility source: aqea-compress commit f093b2d.
```

Abstract

Vector databases have made embedding-based retrieval ubiquitous, yet evaluation practice remains conceptually inconsistent. A single term like "Recall@k" is routinely used to mean either (i) overlap with a baseline model's top-k list, or (ii) retrieval performance against a human or business ground truth. This conflation is comparable to mixing units such as power and energy: the outputs are not commensurate, and optimizing the wrong quantity can increase bias while producing misleading "quality" claims.

We propose a ground-truth-aware terminology standard: every reported metric name must encode the **ground truth (GT)** it is evaluated against. We define canonical terms, a legacy-to-canonical mapping,

and a minimal reproducible evaluation protocol.

1. The problem: identical names, different meanings

In practice, teams deploy vector retrieval but evaluate it with ambiguous metrics. The most common failure mode:

> A metric name is reported without stating what "relevance" means.

Without an explicit ground truth, "Recall@10" has at least two incompatible meanings:

- **Baseline overlap**: "How similar are my results to a baseline embedding model's top-10?"
- **Truth-based retrieval**: "How often do I retrieve items labeled relevant by humans or business scores in the top-10?"

These questions are not interchangeable: optimizing the first reproduces baseline bias; optimizing the second improves customer utility.

1.1 Root causes in real systems (why this persists)

The lack of unambiguous terminology is not an accident. It is an emergent property of how vector retrieval entered production:

A) Cross-discipline term drift (IR vs ML vs Infra)

- **IR heritage**: metrics like Precision/Recall@nDCG are well-defined *given* a relevance definition.
- **ML practice**: labels are expensive, so teams default to proxy objectives (correlation/retention/reconstruction) and reuse IR words.
- **Infra reality**: vector DB adoption is often infrastructure-led; evaluation is treated as "a model detail", not part of the system contract.

B) Missing GT is the enabling failure

Most production pipelines do not maintain an explicit, query-level GT. When GT is absent, teams substitute: - baseline rankings ("pseudo-truth") - weak supervision (click logs) without stated debias assumptions

This produces numbers, but not meaning.

C) Incentives favor ambiguity

- Ambiguous terms ("Recall@10") appear universally comparable, which they are not.
- Precise terms ("Baseline Overlap@10 against model X on candidate pool Y") are correct but less convenient for slides and dashboards.

D) Reproducibility debt

Without explicit GT and candidate pool definitions, results cannot be reproduced, audited, or compared across teams.

1.2 The engineering consequence

Ambiguous metrics cause systematic errors: - **Wrong objective**: optimizing to reproduce baseline bias instead of user utility. - **False confidence**: improvements in a baseline-overlap metric may correlate negatively with human/weighted GT. - **Broken contracts**: teams cannot state what "quality" means operationally (for SLOs, regressions, rollbacks).

2. Definitions

2.1 Entities

- Query set (Q)
- Candidate items (D)
- For each query (q Q), a candidate pool (D_q D)
- A retrieval system produces a ranking (R_s(q)) over (D_q)

2.2 Ground truth (GT)

We use **GT (Ground Truth)** as the canonical ML/IR term. (Earlier drafts used "SoT / Source-of-Truth"; GT is more standard and avoids confusion with "single source of truth".) We define explicit ground-truth types (choose one; do not hide it):

- **GT-H (Human Ground Truth)**: human-provided labels or scores (y(q, d))
- **GT-W (Weighted Ground Truth)**: weighted labels/scores (y_w(q, d)) with documented weighting rationale
- **GT-P (Pairwise Ground Truth)**: pairwise preferences (d_i d_j) for a query (q)
- **GT-L (Log Ground Truth)**: implicit feedback (clicks, dwell time), explicitly treated as biased
- **Baseline-B**: baseline model outputs treated as a reference (not as truth)

2.3 Why baseline is not truth

Baseline outputs can be a valuable engineering reference, but they are not reality. If baseline rankings disagree with human or weighted truth, "high recall to baseline" simply means "high agreement to bias".

2.4 Audit checklist (minimum bar for scientific/engineering validity)

Before reporting any number, you must be able to answer:

1) **What is the task?** (retrieval vs ranking vs pairwise similarity vs reconstruction)

2) **What is the ground truth?** (`GT-H`, `GT-W`, `GT-P`, `GT-L`, or `baseline=B`)

3) **What is the candidate pool definition?** (how is (D_q) constructed?)

4) **What is the metric definition?** (including (k), thresholds, gains)

5) **What is the uncertainty estimate?** (CI across queries, not just a point estimate)

If any of these are missing, the metric is not interpretable.

3. Canonical metric naming standard

3.1 Rule (mandatory)

Every metric must specify its ground truth in the name or annotation:

`Metric@k[GT-*]` or `Metric[baseline=B]`.

Baseline specification (recommended): When the baseline is the original float32 embedding model, prefer the explicit form:

`Metric@k[baseline=original-float32]`

This makes it unambiguous that "baseline" refers to the uncompressed reference, not a different model or a quantized variant.

3.2 Canonical metric families

A) GT-based retrieval metrics (customer-relevant) Use these for product claims and decision-making.

- **nDCG@k[GT-H]**, **nDCG@k[GT-W]**
  - For graded or weighted relevance.
- **Precision@k[GT-*]**, **Recall@k[GT-*]**
  - Only when relevance is well-defined and binary (or has a clear threshold).
- **SetOverlap@k[GT-*]** (GT-defined top-k set overlap)
  - Measures |system_top_k ∩ GT_top_k| / k.
  - Alias: `Agree@k[GT-*]` (legacy), `HitRate@k[GT-*]` (ANN convention when GT is explicit).

B) Baseline-overlap metrics (diagnostic only) Use these for regression testing and stability checks; never market them as "retrieval quality".

- **BO@k (Baseline Overlap @k)**, alias **BRecall@k**:
  - Overlap between system top-k and baseline top-k.
  - Canonical notation: `BO@k[baseline=B]` or `BRecall@k[baseline=B]`.
  - The alias `BRecall@k` preserves familiarity with "Recall" while making the baseline reference explicit.
- **BRA@k (Baseline Rank Agreement @k)**:
  - Rank correlation restricted to top-k (or top-k weighted).

C) Pairwise correlation metrics (STS-style evaluation) Appropriate when the task is naturally defined as pairwise similarity judgments.

- **Spearman[GT-H]**, **Spearman[GT-W]**
- **Kendall[GT-H]**, **Kendall[GT-W]**

D) Reconstruction metrics (compression-intrinsic) These are not retrieval utility metrics. They should never be substituted for GT-based retrieval metrics.

- **ReconError** (e.g., MSE), **ExplainedVar**, etc.

4. Legacy-to-canonical mapping

Legacy term (ambiguous)	Hidden meaning (common)	Canonical replacement
"Recall@10" (no GT)	baseline top-10 overlap	`BO@10[baseline=B]`
"Hits@k"	often baseline top-k overlap in ANN / embedding eval	`BO@k[baseline=B]` (diagnostic) or `Recall@k[GT-*]` (if GT is explicit)
"Accuracy@k" (in retrieval)	either top-k contains a relevant item, or baseline overlap	`SetOverlap@k[GT-*]` (GT-defined top-k) or `BO@k[baseline=B]` (baseline)
"Quality retention"	Spearman vs baseline similarities	`Spearman[baseline=B]`
"Retrieval@k"	mixed	`nDCG@k[GT-*]` or `SetOverlap@k[GT-*]`
"Agreement@k" (unspecified)	unclear	`SetOverlap@k[GT-H]` / `SetOverlap@k[GT-W]`

5. Minimal reproducible protocol (GT-based retrieval)

Step 1 — Define candidate pools (D_q)

Candidate pool design is part of the task definition. Examples: - same language, time window, category constraints - fixed-size hard-negative pool

Requirement: pool construction must not leak the system under test.

Step 2 — Score and rank

For each (q) and (d D_q), compute a score (S_s(q, d)) and rank to obtain (R_s(q)).

Step 3 — Evaluate against GT

Compute at minimum: - `nDCG@k[GT-*]` (graded or weighted truth) - `SetOverlap@k[GT-*]` (GT-defined top-k overlap)

Optionally: - `Precision@k[GT-*]` / `Recall@k[GT-*]` for binary truth

Step 4 — Uncertainty

Report bootstrap confidence intervals across queries (Q) (median + 95% CI).

6. Worked examples (toy)

6.1 Baseline overlap vs GT utility

Assume query (q) with 5 candidates. Baseline top-2 = {A, B}. GT top-2 (human) = {A, C}. System top-2 = {A, C}.

- `BO@2[baseline=B]` = overlap({A, C}, {A, B}) = 1/2
- `SetOverlap@2[GT-H]` = overlap({A, C}, {A, C}) = 2/2

Same system, different conclusions depending on the ground truth.

6.2 STS-style tasks

If your GT is a human similarity score for sentence pairs, a correlation metric is appropriate: `Spearman[GT-H]`. Do not label this as "Recall@k", because there is no retrieval pool.

6.3 Real-system case study: "baseline is already wrong"

This failure mode appears in real products and demos. Consider the public AQEA demo at `https://demo.aqea.ai/` (text similarity, STS12–16 unseen).

Observation A — the baseline already deviates from human truth

The demo exposes the baseline–human correlation:

- `spearmanHumanVsOriginal  0.8486`

Interpreting this as "the baseline is not the same as human GT" is the key point. Whatever proxy you compute *against the baseline* is not a proxy for human utility; it is a proxy for baseline agreement.

Observation B — "Recall@k" in many pipelines is actually baseline overlap

In top-k retrieval UX, the demo reports **Agreement@10** as overlap between baseline top-10 and compressed top-10. This is explicitly a baseline-overlap metric (diagnostic), even if variable names elsewhere might call it "recall".

Canonical naming:

- `BO@10[baseline=B]` (Baseline Overlap @10)

Why this makes Recall@k misleading (the "blind vs one-eyed" effect)

Figure (placeholder): Blind vs one-eyed overlap intuition

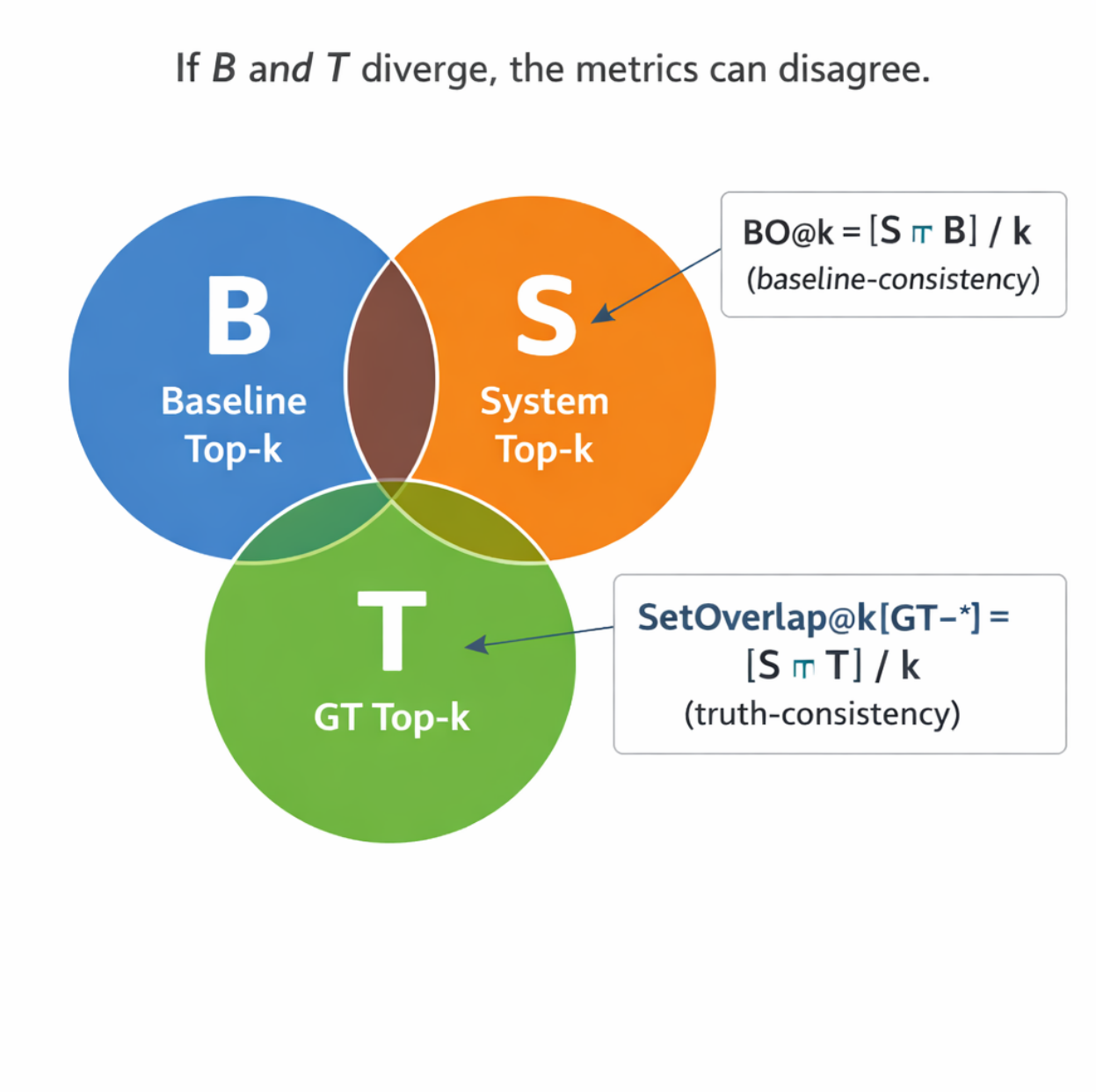

Figure 1: Blind vs one-eyed overlap intuition

Canonical asset: `aqea-technical-report/assets/venn_blind_vs_one_eyed.png` (this bundle includes a copy for PDF builds).

If the baseline is already misaligned with GT, optimizing overlap to baseline is like "measuring yourself against a blindfolded judge while a one-eyed judge stands next to you". A system can legitimately move

closer to GT while becoming *less similar to the baseline*. Without explicit GT labeling, baseline overlap can therefore penalize real improvements.

6.4 Realistic case study (illustrative): Lens improves GT while lowering baseline overlap

Status: illustrative example for clarity (not a quantitative product claim).
Purpose: demonstrate a common real-world situation in which "Recall@k" (when used as baseline overlap) becomes actively misleading.

Setup (typical production retrieval)

- Task: top-k retrieval for a domain application (e.g., e-commerce search or support-ticket routing)
- Candidate pool per query: $|D_q| = 10{,}000$ items (same locale + same product category / same tenant)
- Ground truth: **GT-W** (weighted labels derived from business judgments + curated human review)
- Metrics reported:
  - `nDCG@10[GT-W]` (primary)
  - `SetOverlap@10[GT-W]` (primary)
  - `BO@10[baseline=B]` (diagnostic only)

Two systems

- **Baseline (B):** original transformer embeddings (untuned to the customer's GT)
- **Lens system (L):** a tuned Lens that explicitly optimizes toward the customer's GT

What happens (the key pattern)

The Lens system moves the ranking **toward GT**, which necessarily means it may move **away from baseline**.

Example aggregated results over queries (median; bootstrap CI omitted for brevity):

System	`nDCG@10[GT-W]`	`SetOverlap@10[GT-W]`	`BO@10[baseline=B]`
Baseline (B)	0.420	0.31	1.00
Lens (L)	**0.487**	**0.38**	**0.62**

Interpretation: - The Lens is **better for the customer** (higher GT-based metrics). - A naive "Recall@10" dashboard implemented as "overlap vs baseline top-10" would report a drop (from 100% to 62%) and would incorrectly flag this as a regression.

Why this is inevitable when GT differs from baseline

If baseline and GT disagree, improving against GT requires changing the ranking. Any metric that treats baseline as truth will penalize that change.

Canonical wording: - "`BO@10[baseline=B]` decreased" is a **diagnostic observation**. - "`nDCG@10[GT-W]` increased" is the **actual system improvement**.

The "blind vs one-eyed" framing (engineering version)

When baseline itself is misaligned with GT, optimizing overlap to baseline is like comparing yourself to a blind evaluator while the one-eyed evaluator (GT) stands next to you. The only scientifically valid comparator is the explicitly stated GT.

6.5 Observed pattern (domain): Legal search

Status: domain scenario description (no new quantitative claims).
Purpose: show why baseline-overlap metrics can label true improvements as regressions in legal retrieval.

Setup

- Task: retrieval for legal research / case-law search (top-k results used by a human)
- Candidate pool (D_q): jurisdiction + time-window constrained; excludes duplicates; includes hard negatives (near-topic but legally irrelevant)
- Ground truth: **GT-W** derived from:
  - legal expert judgments (relevance + authority)
  - weights reflecting business value (e.g., binding precedent > persuasive precedent; recency; jurisdiction match)
- Metrics:
  - `nDCG@10[GT-W]` (primary)
  - `SetOverlap@10[GT-W]` (primary)
  - `BO@10[baseline=B]` (diagnostic only)

Why baseline diverges from GT in legal search

General embedding models often correlate with "semantic similarity of wording", but legal relevance depends on additional structure: - authority and jurisdiction - procedural posture - exceptions and narrowly-scoped holdings

As a result, baseline rankings can be systematically misaligned with expert-weighted GT.

Expected metric behavior when a Lens improves the system

If a Lens is tuned toward GT-W, it will: - move binding/authoritative sources upward (GT utility ↑) - demote semantically-similar-but-legally-wrong matches (GT utility ↑)

This necessarily changes the ranking relative to the baseline: - `BO@10[baseline=B]` can decrease even while `nDCG@10[GT-W]` increases.

Practical guidance (for teams)

- Use baseline overlap only as a regression diagnostic for "did we change behavior?"
- Use GT-based metrics as the acceptance gate for "did we improve legal utility?"

6.6 Observed pattern (domain): E-commerce search and recommendations

Status: domain scenario description (no new quantitative claims).
Purpose: show why "baseline agreement" is not equivalent to business relevance in product search.

Setup

- Task: product search / recommendations (top-k drives clicks and purchases)
- Candidate pool (D_q): category-constrained + in-stock filter + price band; hard negatives included (nearby brands, similar descriptions)
- Ground truth: **GT-W** derived from:
  - human merchandising judgments
  - conversion-weighted implicit signals (click→cart→purchase funnel), explicitly modeled as weighted truth
- Metrics:
  - `nDCG@10[GT-W]` (primary)
  - `SetOverlap@10[GT-W]` (primary)

– `BO@10[baseline=B]` (diagnostic only)

Why baseline diverges from GT in e-commerce

The "most semantically similar description" is often not the "most relevant result": - availability, price sensitivity, brand preferences - intent ambiguity ("apple" the fruit vs Apple products) - business constraints (margin, sponsored items, policy filters)

So a general baseline embedding ranking is not an GT for business relevance.

Expected metric behavior when Lens tuning works

When a Lens aligns retrieval with GT-W (business utility), it can: - increase `nDCG@10[GT-W]` and `SetOverlap@10[GT-W]` - decrease `BO@10[baseline=B]` because the system intentionally departs from baseline semantics

Practical guidance (for teams)

- Treat `BO@k[baseline=B]` as "change magnitude", not as "quality".
- Do not block launches on baseline-overlap drops if GT-based metrics improve.

7. Anti-patterns (explicitly forbidden)

- Reporting `Recall@k` without specifying GT.
- Treating baseline top-k as truth while presenting it as "retrieval quality".
- Mixing intrinsic (structure retention) and extrinsic (truth-based utility) in the same headline without separation.

Stronger rule: Reporting a metric without `[GT-*]` or `[baseline=B]` annotation is **scientifically invalid** and should be rejected in peer review and in engineering PR review.

8. Reporting template (recommended)

Every result table must include: - **Task** (retrieval / ranking / correlation) - **GT type** (`GT-H`, `GT-W`, `GT-P`, `GT-L`, or `baseline=B`) - Dataset + sampling protocol (candidate pool definition) - Metric definitions and (k) - Uncertainty (CI)

8.1 Adoption path (how this becomes a standard)

To make terminology stick in real organizations, treat it as a contract:

- **Dashboards**: every metric label includes `[GT-*]` or `[baseline=*]`.
- **UI/Charts**: always render the full label (e.g., `BO@10[baseline=original-float32]`), not a shortened internal name.
- **PR reviews**: forbid ambiguous "Recall@k" in code/docs unless GT is specified.
- **Benchmarks**: separate sections for baseline diagnostics vs GT-based utility.
- **Public claims**: only GT-based metrics qualify as "quality"; baseline overlap is explicitly "diagnostic".

8.2 Machine-readable metric header (schema)

Terminology becomes industry-grade when it is machine-checkable. We recommend attaching a minimal structured header (YAML/JSON) to every metric table row, dashboard series, or benchmark output.

Example (YAML):

```yaml
task_type: retrieval            # retrieval | ranking | correlation | reconstruction
gt_type: GT-W                   # GT-H | GT-W | GT-P | GT-L | baseline
baseline_id: original-float32   # required if gt_type=baseline
```

```yaml
candidate_pool_id: pool_v1
candidate_pool_desc: "tenant=acme, locale=en, category=shoes, hard-negatives"
metric: nDCG                    # nDCG | BO | BRecall | SetOverlap | Precision | Recall | Spearman | Ke
k: 10
ci_method: bootstrap
ci_level: 0.95
```

This enables linting, reproducibility, and cross-team comparison without re-litigating hidden assumptions.

8.3 Metric linter (engineering bridge)

To bridge science → production, teams should enforce a simple CI rule:

- Reject metrics labeled as `Recall@k` / `Hits@k` / `Accuracy@k` unless they carry an explicit annotation: `[GT-*]` or `[baseline=*]`.
- Reject any dashboard/benchmark line missing: task type, GT type, and candidate pool definition.

This converts terminology from a style guide into an operational contract.

9. References

- AQEA Technical Report (primary reference): DOI `10.5281/zenodo.18138436` (`https://doi.org/10.5281/zenodo.18`

10. Related work (why this matters across subfields)

This terminology issue spans multiple sub-communities:

10.1 Retrieval benchmarks (explicit GT)

- **BEIR** defines a heterogeneous set of IR datasets with explicit relevance labels and standard IR metrics (e.g. nDCG@k).
  Citation: "BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models" (arXiv: `2104.08663`) — `https://arxiv.org/abs/2104.08663`.

10.2 Embedding benchmarks (mixed metric surface)

- **MTEB** benchmarks embedding models across tasks including retrieval and commonly reports top-k metrics in retrieval tasks.
  Citation: "MTEB: Massive Text Embedding Benchmark" (arXiv: `2210.07316`) — `https://arxiv.org/abs/2210.0731`

10.3 Multi-resolution embeddings (how naming needs to stay GT-aware)

- **Matryoshka Representation Learning** targets multi-resolution embeddings and motivates evaluation across truncation levels.
  Citation: "Matryoshka Representation Learning" (arXiv: `2205.13147`) — `https://arxiv.org/abs/2205.13147`.

10.4 ANN compression / quantization (baseline-defined "recall")

Approximate nearest neighbor (ANN) compression papers often define "Recall@k" as agreement to **nearest neighbors in the original vector space**. Under this whitepaper's standard, that is valid but must be named as baseline-based:- Canonical: `BO@k[baseline=original-float32]` or `BRecall@k[baseline=original-float32]`

Representative references:

- "Product Quantization for Nearest Neighbor Search" (HAL/Inria landing page): `https://hal.inria.fr/inria-00514`

- “Optimized Product Quantization for Approximate Nearest Neighbor Search” (DBLP entry): `https://dblp.org/rec/conf/cvpr/GeHK013`

11. Call to action

We invite: - benchmark maintainers (e.g., MTEB/BEIR-style leaderboards), - vector database vendors, - and evaluation tooling authors

to adopt ground-truth-aware metric labels (`[GT-*]` / `[baseline=*]`) in dashboards, docs, and papers.