Semantic Data Normal Forms: Extending Normalization Theory to Vector Embedding Spaces

Anonymous SEM 2026 Submission

Abstract

AI-native data systems encode schema meaning with vector embeddings, creating integrity challenges: stochastic instability, semantic drift, and context mixing. We introduce Semantic Data Normal Forms (SDNF), a framework that extends classical normalization to embedding spaces via a Semantic Relational Schema (SRS) and seven normal forms enforcing stability, alias resolution, context isolation, drift bounds, evidence completeness, role consistency, and partition orthogonality. Under explicit assumptions we present a conditional cross-context safety result and a governance descent argument that supports schema convergence in practice. A compact governance pipeline implements SDNF with evidence aggregation and lineage. Experiments on payments payloads demonstrate $\sim$35–40% schema consolidation, high merge precision, and low measured context leakage under the evaluated settings. SDNF provides an auditable, evidence-driven approach to schema governance for embedding-based systems; extension to other domains is left to future work.

1 Introduction

Modern data schemas act as semantic interfaces between AI models, services, and databases. Unlike symbolic schemas, AI-native systems use vector embeddings to represent meaning; similarity is determined by high-dimensional proximity. This shift introduces three failure modes: *Stochastic Instability* (repeated encoding of the same concept yields non-identical vectors due to model nondeterminism), *Semantic Drift* (model updates or data evolution shift concept representations over time), and *Context Mixing* (distinct domains overlap in embedding space, e.g., "card" as payment vs. playing card). Classical normalization (1NF–BCNF) addresses structural redundancy but not semantic corruption. SDNF extends normalization theory into semantic space, enabling progressive refinement from payload-derived schemas to stable canonical schemas with auditable lineage.

Contributions:

(a) A formal SRS model with multi-level embeddings and context projections.

(b) Seven Semantic Normal Forms (EENF, AANF, CMNF, DBNF, ECNF, RRNF, PONF).

(c) Conditional theoretical results under explicit assumptions.

(d) A governance pipeline implementation with evidence aggregation and lineage.

(e) Experimental validation on payments payloads showing substantial consolidation and high precision.

2 Semantic Relational Schema (SRS) Model

2.1 Formal definition

An SRS is a tuple $\mathrm{SRS} = (E, A, R, \mathrm{Emb}, C, L)$ where:

- E: set of entities (canonical concepts).
- A: attributes with metadata (type, regex, provenance, aliases).
- R: semantic relations (source, target, confidence, role).
- $\mathrm{Emb}(p, c)$: embedding function $P \times C \to \mathbb{R}^d$.
- C: semantic contexts (e.g., Payments, Risk).
- L: append-only lineage log of evolution events with evidence.

2.2 Multi-level embedding decomposition

For primitive p in context c:

$$\begin{aligned}\mathrm{Emb}(p,c) = &\big[\,\mathrm{Emb}_{\text{fine}}(p,c) \parallel \\ &\mathrm{Emb}_{\text{abstract}}(p) \parallel \mathrm{Emb}_{\text{contextual}}(p,c)\big].\end{aligned} \tag{1}$$

Each subvector is ℓ_2-normalized before concatenation. This decomposition separates lexical, context-independent, and domain-modulated semantics and supports progressive convergence as evidence accumulates.

Fine-level embedding $\mathrm{Emb}_{\text{fine}}(p,c)$ captures the lexical or token-level semantics of the primitive p within context c. *Explanation:* It is similar to precise word-level meaning, sensitive to spelling, morphology, and local usage.

Abstract embedding $\mathrm{Emb}_{\text{abstract}}(p)$ represents the context-independent, generalized semantics of the primitive p. *Explanation:* This is the distilled, higher-level meaning of the word or concept, invariant across domains.

Contextual embedding $\mathrm{Emb}_{\text{contextual}}(p,c)$ encodes the domain-modulated semantics of the primitive p in context c. *Explanation:* This adjusts the meaning depending on the domain — e.g., "operation" as a transaction in Payments vs. a surgery in Medical.

Together, these components ensure that embeddings can be decomposed into fine, abstract, and contextual parts, encouraging separation of concerns and reducing semantic overlap across domains.

2.3 Context projection operator

$$\begin{aligned}\|\,\mathrm{Proj}_c\,\| &\le \kappa, \\ \|\,\mathrm{Proj}_c^2 - \mathrm{Proj}_c\,\| &\le \delta, \\ \|\,\mathrm{Proj}_c(x+\eta) - \mathrm{Proj}_c(x)\| &\le \kappa\|\eta\|.\end{aligned} \tag{2}$$

For each context c, define $\mathrm{Proj}_c : \mathbb{R}^d \to \mathbb{R}^d$ with the following properties:

2.3.1 Bounded amplification

$$\|\,\mathrm{Proj}_c\,\| \le \kappa$$

The projection should not magnify the size of any vector beyond a safe bound κ. *Note:* Like a volume knob that can amplify but never exceed a set limit.

2.3.2 Approximate idempotence

$$\|\,\mathrm{Proj}_c^2 - \mathrm{Proj}_c\,\| \le \delta$$

Applying the projection twice should be almost the same as applying it once, with only a small error δ. *Note:* Similar to putting on tinted glasses — wearing them twice does not change the tint further.

2.3.3 Stability under perturbations

$$\|\,\mathrm{Proj}_c(x+\eta) - \mathrm{Proj}_c(x)\| \le \kappa\|\eta\|$$

Small changes in the input (η) should only cause proportionally small changes in the projected output. *Note:* If we nudge a picture slightly, the filtered version shifts only a little, not drastically.

Practical realizations include PCA/ridge linear maps (W_c) and attention-based diagonal modulation. These projections encourage near-orthogonality of identical tokens across distinct contexts, ensuring that meanings remain separated across domains.

2.4 SRS metric and evidence inner product

Definition 1: SRS metric For primitives p_1, p_2 in context c:

$$\begin{aligned}&d_{\mathrm{SRS}}\big((p_1,c),(p_2,c)\big) = \\ &\big\|\,\mathrm{Proj}_c(\mathrm{Emb}(p_1,c)) - \mathrm{Proj}_c(\mathrm{Emb}(p_2,c))\big\|_2.\end{aligned} \tag{3}$$

Explanation: The SRS metric measures the semantic distance between two primitives after projecting them into the same context. Consider comparing how far apart two words are when viewed them through the same "contextual lens." If the distance is small, they carry similar meaning in that domain; if large, they diverge semantically.

Definition 2: Evidence-weighted inner product For entities e_1, e_2 with evidence sets $\mathrm{Ev}_1, \mathrm{Ev}_2$:

$$\langle e_1, e_2\rangle_E := \sum_i w_i(\mathrm{Ev}_i)\,\langle \mathrm{Emb}(e_1), \mathrm{Emb}(e_2)\rangle,$$

with normalized positive weights w_i.

Explanation: The evidence-weighted inner product compares two entities by weighting their similarity according to supporting evidence. Considering each piece of evidence as a vote, with stronger or more reliable signals carrying higher weight. This ensures that similarity is not judged solely by embeddings, but by how well evidence supports the

comparison. For finite or bounded evidence sets, this weighted inner product induces a pre-Hilbert structure on SRS, meaning it behaves like a well-defined inner product space suitable for rigorous analysis. A full proof follows directly from the inner product axioms and is omitted for brevity.

Together, the SRS metric and evidence-weighted inner product provide both a geometric notion of distance and an evidence-aware notion of similarity, ensuring semantic comparisons are both mathematically sound and contextually grounded.

3 Semantic Data Normal Forms (SDNF)

Each NF targets a semantic anomaly with a formal constraint, validation metric, and remediation.

3.1 EENF — Entity Embedding Normal Form (stability)

Constraint: For entity e in context c,

$$\mathrm{Var}\left(\{\mathrm{Emb}^g(e,c)\}_{g=1}^{G}\right) \le \tau_{\mathrm{EENF}}.$$

Explanation: The variance across multiple regenerated embeddings of the same entity should remain below a stability threshold τ_{EENF}. If embeddings fluctuate too much, the representation is unstable and may corrupt downstream semantics.

Action: Quarantine unstable entities; increase regenerations (G), apply deterministic seeding, or select a more stable encoder.

3.2 AANF — Attribute Alias Normal Form (alias detection)

Constraint Attributes a, b are considered aliases if:

$$\begin{aligned} &\cos\big(\mathrm{Emb}(a,c_a), \mathrm{Emb}(b,c_b)\big) \ge \tau_{\mathrm{AANF}}, \\ &\mathrm{ontology\_root}(a) = \mathrm{ontology\_root}(b), \\ &\mathrm{EvidenceSet}(a,b) \ge m_{\min}. \end{aligned} \tag{4}$$

Explanation: Two attributes are flagged as aliases when three conditions hold: 1. Their embeddings are highly similar (cosine similarity above threshold τ_{AANF}). 2. They share the same ontology root, meaning they belong to the same conceptual category. 3. There is sufficient supporting evidence ($m_{\min}$ or more).

Example: Discovering that "DOB" and "Date of Birth" are different labels for the same concept — the math checks similarity, the ontology confirms they belong to the same family, and evidence ensures it is not a coincidence.

Action Auto-merge when ECNF evidence thresholds are met; otherwise defer.

Explanation: If enough evidence supports the alias relationship, the system automatically merges the attributes to prevent redundancy. If evidence is insufficient, the merge is deferred until more proof accumulates.

3.3 CMNF — Context Modulation Normal Form (context isolation)

Constraint For primitive p in contexts c_1, c_2:

$$\begin{aligned} &\Big\langle \mathrm{Proj}_{c_1}\big(\mathrm{Emb}(p,c_1)\big), \\ &\mathrm{Proj}_{c_2}\big(\mathrm{Emb}(p,c_2)\big)\Big\rangle \\ &\qquad \le \tau_{\mathrm{CMNF}}. \end{aligned} \tag{5}$$

Explanation: The inner product between the projections of the same primitive in two distinct contexts must remain below a threshold τ_{CMNF}. This ensures that meanings do not leak across domains.

Action Retrain or re-estimate projections; flag contamination.

Explanation: If cross-context overlap exceeds the threshold, the projection matrices must be retrained or adjusted. Entities showing contamination are flagged for review.

3.4 DBNF — Drift-Bounded Normal Form (drift control)

Constraint For entity e across model versions v_1, v_2:

$$\big\|\,\mathrm{Emb}(e,v_1) - \mathrm{Emb}(e,v_2)\big\| \le \tau_{\mathrm{DBNF}}.$$

Explanation: The distance between embeddings of the same entity across different model versions must remain below a drift threshold τ_{DBNF}. This ensures that updates to the model do not distort the meaning of entities beyond acceptable bounds. *Example:* Updating a dictionary edition — the definition of "bank" should not suddenly shift from "financial institution" to "river edge" unless explicitly intended.

Action Fork versions when drift exceeds threshold; preserve historical semantics.

Explanation: If drift surpasses the threshold, the system forks the model version to preserve historical semantics. This prevents silent corruption of meaning and allows both old and new interpretations to coexist.

3.5 ECNF — Evidence Completeness Normal Form (explainability)

Constraint: Operation O permitted only if

$$\begin{aligned} \text{EvidenceSet}(O) &\geq m_{\min}, \\ \text{aggregate\_score}\big(\text{EvidenceSet}(O)\big) &\geq \gamma. \end{aligned} \tag{6}$$

Action: Auto-merge if satisfied; otherwise defer to human review.

Practical note: value evidence is often missing (27.6% in our dataset); the governance pipeline falls back to hybrid evidence (name + ontology + VSS + shape + embeddings) to preserve precision while recovering recall (see §VII and Appendix C).

3.6 RRNF & PONF (roles and partition orthogonality)

RRNF — Role-Respecting Normal Form Constraint: Enforce role consistency in relations to prevent invalid transitive inferences.

Explanation: Each entity must preserve its assigned role across relations. Violations occur when roles are swapped or blurred, leading to incorrect logical deductions.

Example: Ensuring that in a family tree, a "parent" never suddenly becomes a "child" in another branch. Consistency prevents nonsensical transitive inferences.

PONF — Partition Orthogonality Normal Form Constraint: Maintain orthogonality of semantic partitions; re-partition when overlap exceeds threshold τ_{PONF}.

$$\text{Overlap}(P_i, P_j) \leq \tau_{\text{PONF}}, \quad \forall i \neq j$$

Explanation: Semantic partitions (clusters of meaning) must remain distinct. If overlap between partitions grows beyond τ_{PONF}, the system must re-partition to restore orthogonality. *Example:* Keeping different departments in an organization separate — if Finance and HR roles start overlapping too much, responsibilities must be redefined to avoid confusion.

Action - For RRNF: Validate and enforce role consistency; quarantine or correct relations that violate role assignments. - For PONF: Monitor partition overlaps; trigger re-partitioning when thresholds are exceeded to preserve semantic clarity.

4 Mathematical Analysis

4.1 Assumptions (explicit)

Projection stability (A1): Proj_c are **bounded**, approximately **idempotent**, and **stable** as defined in §II-C.

Boundedness: Projections act like filters — they never amplify signals beyond a safe limit. *Idempotence:* Applying the projection twice is essentially the same as once, ensuring consistency. *Stability:* Small input changes only cause proportionally small output changes,

Embedding Lipschitz (A2): For finite contexts, Emb is Lipschitz (upper bound) and injective on the finite vocabulary, enabling conditional bi-Lipschitz statements.

Lipschitz continuity: Distances in meaning are preserved in a controlled way — similar tokens remain close, dissimilar ones remain apart. *Injectivity:* Each token maps to a unique embedding, avoiding collisions. Example: assigning unique ID cards to students — no two share the same card, and their relative closeness is preserved.

Subspace model (A3): Embeddings for distinct contexts concentrate in approximately low-dimensional subspaces; concentration inequalities apply. *Intuition:* Each domain occupies its own compact "semantic room." Finance terms cluster in one corner, Medical terms in another, with most points tightly packed and only a few drifting out.

Governance descent (A4): The governance pipeline is designed to monotonically decrease a redundancy metric $L(S)$ in expectation; discrete decisions are constrained to avoid increases in $L(S)$. *Intuition:* Governance acts like a catalog clean-up crew — every step reduces duplication and overlap, and no action is allowed to add new duplicates.

These assumptions are stated to make the subsequent theorems precise; proofs and technical bounds are in Appendix A–B.

4.2 Conditional theorems (statements and sketches)

Theorem 1 (CMNF Cross-Context Safety). Under A1–A4, if CMNF holds for contexts c_1, c_2 (i.e., for all p, inner products $\leq \tau_{\text{CMNF}}$), then under the subspace model the probability of cross-context retrieval error is bounded by an explicit function $f(\tau_{\text{CMNF}}; d, n)$ that decays as $\tau_{\text{CMNF}} \to 0$ and with increasing ambient dimension d. Proof sketch; full proof under the stated assumptions is in Appendix A.

Intuition: This theorem ensures that when contexts are kept sufficiently isolated, the risk of mixing or confusing meanings across domains becomes negligible, especially in higher dimensions.

Theorem 2 (DBNF Drift Fork Necessity). Under A1–A4, if $\|\operatorname{Emb}(e, v_1) - \operatorname{Emb}(e, v_2)\| > \tau_{\mathrm{DBNF}}$, then for any downstream operator O that is L_O-Lipschitz,

$$\|O(\operatorname{Emb}(e, v_1)) - O(\operatorname{Emb}(e, v_2))\| \geq L_O \cdot \operatorname{drift} - K \quad (7)$$

When the right-hand side exceeds an application threshold, forking is required to preserve semantic consistency. Proof sketch; full proof in Appendix A.

Intuition: This theorem formalizes when semantic drift across model versions becomes too large to ignore, showing that a fork is necessary to maintain consistent downstream behavior.

Theorem 3 (SRS Completeness Under SDNF). Under A1–A4, for finite contexts and under boundedness and closure conditions, $(\mathrm{SRS}, d_{\mathrm{SRS}})$ is complete: every Cauchy sequence converges within SRS. Moreover, under additional technical bounds on governance updates (Appendix B), the governance pipeline converges to a canonical fixed point. Proof sketch; full proof in Appendix A and Appendix B.

Intuition: This theorem guarantees that the semantic space is mathematically well-formed — sequences of embeddings converge, and governance processes stabilize to a consistent schema.

Corollary 3.1 (Fixed-Point Convergence). Under A1–A4 and the governance descent bounds in Appendix B, the governance pipeline converges to a canonical schema $S^\star$ with $T(S^\star) = S^\star$ (see Appendix B).

Intuition: This corollary confirms that governance does not wander indefinitely but settles into a stable fixed schema.

Corollary 3.2 (Conditional Bi-Lipschitz Embedding). Under A2 (finite contexts) Emb is bi-Lipschitz up to controlled distortion; constants depend on κ and τ_{AANF} (formal statement and bounds in Appendix A).

Intuition: This corollary shows that embeddings preserve both closeness and separation of tokens within finite contexts, ensuring semantic distances remain meaningful under controlled distortion.

4.3 Governance descent (replacement for unconditional contraction)

The governance pipeline is a composition of continuous centroid updates and discrete merge/fork decisions. Continuous updates are Lipschitz-bounded; discrete decisions are implemented to avoid increases in a redundancy metric $L(S)$. Under explicit bounds on discrete update magnitudes (Appendix B), repeated application of the pipeline yields monotone descent of $L(S)$ and convergence to a fixed point. The Banach contraction formulation is presented conditionally in Appendix B where required Lipschitz constants are made explicit.

Intuition: - **Lipschitz-bounded:** Each update changes the system in a controlled way — no sudden jumps, like adjusting a thermostat where small tweaks only cause proportionally small temperature shifts. - **Banach contraction:** Repeatedly applying the pipeline steadily pulls the system closer to a single stable state, like folding a piece of paper in half again and again — it always converges toward one point. *Example:* Governance descent works like a navigation app recalculating routes: continuous updates gently nudge you closer to the destination, while discrete reroutes (merge/fork) avoid detours. Over time, you always end up at the fixed destination without wandering endlessly.

5 Governance Pipeline

Stages: Payload ingestion → Schema derivation → Embedding generation (multi-level) → Context projection → SDNF validation (EENF → AANF → CMNF → DBNF → ECNF → RRNF → PONF) → Evidence aggregation → Decision (merge/fork/defer/quarantine) → Lineage recording.

Evidence signals: embedding similarities (fine/abstract/contextual), name token overlap, ontology match, value semantic signature (VSS), shape tokens, co-occurrence statistics, regex heuristics. Aggregate scoring uses configurable weights; ECNF enforces minimum distinct signals and score thresholds.

Operational rules: Auto-merge only when AANF + ECNF satisfied; fork on DBNF violations; quarantine on EENF failures; defer otherwise. Lineage entries record evidence ids, timestamps, actor, and decision rationale.

6 Experiments

6.1 Setup (compact)

Platform: Python 3.10; Sentence-Transformers `all-MiniLM-L6-v2` (model version in Appendix C); HNSW (hnswlib M=32, ef=50). Data: payments domain (`INAmex.json`, `PPVisa.json`, ∼50 payloads). Metrics: schema consolidation (% reduction), merge precision/recall (±CI), context leakage rate, drift detection accuracy.

6.2 EENF mitigation and reproducibility

We observed embedding variance from nondeterministic encoders. To quantify and mitigate this we use regeneration trials (G) and report sensitivity results. Experiments use the model and `hnswlib` versions listed in Appendix C and fixed random seeds (Appendix C). For production we recommend deterministic seeding or model choices that support reproducible encodings. In our ablation, increasing regenerations from G=10 to G=20 reduced mean embedding variance by ∼40% at the cost of ∼2× embedding compute; per-entity quarantine counts are in Appendix C.

6.3 Key results (compact table)

6.4 Ablation highlights (brief)

No ECNF: precision drops to 0.86 (false merges increase).
No CMNF: context leakage rises to ≈ 9%.
No DBNF: drift events undetected, leading to semantic corruption.
Evidence missingness: 27.6% of attributes lack value evidence; hybrid mode (name+ontology+VSS+shape+embeddings) recovers recall while preserving precision (details in Appendix C).

6.5 Example merge (illustrative)

Merging `acct_num` → `PrimaryAccountNumber`: cosine sim 0.99; ontology ISO8583 PAN; co-occurrence 82%; regex match `^[0-9]{13,19}$`; aggregate score $0.92 \geq \gamma \rightarrow$ auto-merge; lineage recorded.

7 Related Work

Classical normalization (Codd, 1970; Kent, 1979) addresses structural redundancy; SDNF extends normalization to vector semantics. Schema matching and ontology embeddings (Rahm and Bernstein, 2001) inform alias detection; vector DB and retrieval literature (Malkov and Yashunin, 2018; Johnson et al., 2019) inform indexing and nearest-neighbor behavior. Representation drift studies motivate DBNF. Metric space and concentration results (Heinonen, 2001; Boucheron et al., 2013) underpin conditional probabilistic bounds; fixed-point theory (Banach, 1922) motivates convergence arguments (all used under stated assumptions).

8 Conclusion

We presented SDNF, a practical framework for governing semantic schemas in embedding-based systems. By combining a formal SRS model, seven semantic normal forms, and an evidence-driven governance pipeline with lineage, SDNF reduces redundancy and mitigates semantic anomalies in practice. Theoretical claims are stated conditionally under explicit assumptions; empirical validation on payments payloads shows substantial consolidation and high precision. Further work will broaden domain validation and refine formal convergence under weaker assumptions; proofs and extended experiments are provided in the appendix.

Limitations

Domain scope: experiments are on payments payloads; claims about other domains are conditional and require further validation.
Nondeterminism: encoder nondeterminism affects EENF; mitigation via G and deterministic seeding is recommended.
Evidence missingness: hybrid fallback is effective but depends on domain heuristics (`ontology_root`).
Scalability: current experiments are modest scale; scalable partitioning strategies are future work.
Reproducibility: model names/versions, seeds, `hnswlib` versions, and full experiment scripts are provided in Appendix C.

References

Stefan Banach. 1922. Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales. *Fundamenta Mathematicae*, 3:133–181.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. 2013. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press.

E. F. Codd. 1970. A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387.

Juha Heinonen. 2001. *Lectures on Analysis on Metric Spaces*. Springer.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

William Kent. 1979. *Data and Reality: Basic Assumptions in Data Processing Reconsidered*. North-Holland.

Yury A. Malkov and Dmitry A. Yashunin. 2018. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):824–836.

Erhard Rahm and Philip A. Bernstein. 2001. A survey of approaches to automatic schema matching. *VLDB Journal*, 10(4):334–350.

Semantic Data Normal Forms GitHub Repo

The entire work and results are all available in a public GitHub repository under the MIT license: pvchaitu/SemanticNormalForms.

Appendix A: Formal Lemmas and Proofs (PCA / Lipschitz Setting)

This appendix gives formal statements and proofs for Theorems 1–3 from the main text in a simplified, rigorous setting. We work under the PCA/subspace model and Lipschitz/bi-Lipschitz operator assumptions called out below. All vectors are real and finite-dimensional. Notation: $\|\cdot\|$ denotes the Euclidean norm, $\langle\cdot,\cdot\rangle$ the Euclidean inner product, and $\mathbb{S}^{d-1}$ the unit sphere in $\mathbb{R}^d$.

A.1 Setup and standing assumptions

(A1) Finite ambient dimension. Embeddings live in $\mathbb{R}^d$ with fixed $d \in \mathbb{N}$. *Intuition:* Fixing dimension ensures distances and angles are well-defined; it makes the geometry tractable.

(A2) PCA/subspace model for contexts. For each context c there exists a linear subspace $U_c \subset \mathbb{R}^d$ of dimension $r_c \le d$ and an orthogonal projector P_{U_c} onto U_c. *Intuition:* Each context occupies its own "semantic room," and projections are the doorway into that room.

(A3) Subspace separation. For two contexts c_1, c_2 define the principal angle $\theta \in [0, \pi/2]$ between subspaces U_{c_1} and U_{c_2} by

$$\cos\theta := \sup_{u \in U_{c_1} \cap \mathbb{S}^{d-1},\, v \in U_{c_2} \cap \mathbb{S}^{d-1}} \langle u, v\rangle.$$

Intuition: θ measures how distinct two contexts are — small angles mean overlap, large angles mean clear separation.

(A4) Random embedding model. For a fixed primitive p, its context-projected embedding in context c is modeled as $x_c = P_{U_c} z_c$ where z_c is a random vector with isotropic concentration. *Intuition:* Embeddings are treated like random samples that concentrate around their context subspace, giving probabilistic control.

(A5) Downstream operator regularity. A downstream operator $O : \mathbb{R}^d \to \mathbb{R}^m$ is assumed bi-Lipschitz:

$$\ell \|x - y\| \le \|O(x) - O(y)\| \le L \|x - y\|.$$

Intuition: Downstream operators stretch or compress distances in a controlled way, never collapsing distinct points or exploding small differences.

A.2 Lemmas on projections and subspace angles

Lemma A.1 (Inner product bound from subspace angle). For any unit vectors $u \in U \cap \mathbb{S}^{d-1}$ and $v \in V \cap \mathbb{S}^{d-1}$, $\langle u, v\rangle \le \cos\theta$. *Intuition:* The maximum similarity between vectors from two subspaces is capped by the cosine of their separating angle.

Lemma A.2 (Projection norm and idempotence). Let P_U be the orthogonal projector onto subspace U. Then $\|P_U\| = 1$ and $P_U^2 = P_U$. *Intuition:* True projections are perfectly stable filters; approximations stay close, with bounded error.

A.3 Concentration of inner products (Gaussian model)

Lemma A.3 (Gaussian inner product concentration). For Gaussian vectors x, y supported on subspaces U, V,

$$\Pr\big(\langle u, v\rangle \ge t\big) \le 2\exp\big(-c\, d_{\text{eff}}\, t^2\big).$$

Intuition: Random vectors from different subspaces almost never align too closely; the chance decays exponentially with dimension.

A.4 Theorem 1 (CMNF Cross-Context Safety)

Theorem A.1. Under (A1)–(A4), for contexts c_1, c_2 with angle θ,

$$\Pr\left(\left\langle \frac{x_{c_1}}{\|x_{c_1}\|}, \frac{x_{c_2}}{\|x_{c_2}\|} \right\rangle \geq \cos\theta + t\right) \leq 2\exp\left(-c\, d_{\text{eff}}\, t^2\right). \tag{8}$$

Intuition: If contexts are sufficiently separated, the probability of confusing them in retrieval drops exponentially with dimension.

A.5 Theorem 2 (DBNF Drift Fork Necessity)

Theorem A.2. Under (A1)–(A5), if $\|x_1 - x_2\| > \tau_{\text{DBNF}}$, then

$$\|O(x_1) - O(x_2)\| \geq \ell\, \|x_1 - x_2\| > \ell\, \tau_{\text{DBNF}}.$$

Intuition: When embeddings drift beyond tolerance, downstream operators magnify the difference, forcing a fork to preserve consistency.

A.6 Theorem 3 (SRS completeness and governance convergence)

Theorem A.3 (SRS completeness). Under (A1)–(A3), $(\mathcal{X}, d_{\text{SRS}})$ is a complete metric space. *Intuition:* The semantic space is mathematically well-behaved — sequences converge, ensuring stability of meaning.

Proposition A.4 (Governance descent and convergence). If $L(S_t)$ is monotone nonincreasing and bounded below, then S_t converges to a fixed point. *Intuition:* Governance acts like a clean-up process: each step reduces redundancy, and the system settles into a stable schema.

Appendix B: Contraction Bounds and Uniqueness

Suppose each pipeline step satisfies a Lipschitz bound with constant $\alpha < 1$ on the metric space $(\mathcal{X}, d_{\text{SRS}})$. Then Banach's fixed-point theorem ensures uniqueness of the limit schema $S^\star$. *Intuition:* Contraction guarantees that repeated updates pull the system closer to one unique stable point, like folding paper in half repeatedly until it collapses to a single crease.

Appendix C: Practical Calibration Notes

Constants c, ℓ, κ are application-dependent. *Intuition:* These parameters act like "knobs" practitioners can tune empirically — effective dimension for concentration, sensitivity constants for drift, and redundancy penalties for governance.