Efficient Logical Reasoning with Hyperdimensional Computing Through Interference-Canceled Hybrid Encoding

Anonymous authors
Paper under double-blind review

Abstract

Hyperdimensional Computing or Vector Symbolic Architectures (HDC/VSA) offer a middle ground between symbolic and neural reasoning, combining compositional structure with efficient vector operations, yet their application to relational reasoning has hit a wall: interference in role-filler bindings degrades the signal-to-noise ratio as predicate arity increases: an unavoidable $1/\sqrt{k}$ decay that makes reliable inference difficult beyond binary or ternary relations. We address this through query-aware decoding rather than a better encoding alone. Our hybrid encoding combines predicate binding with positional permutations, preserving distinguishability while enabling invertible argument recovery, and successive interference cancellation (SIC) exploits known query constraints to remove interference before cleanup. Retrieval similarity improves from 0.31 to 0.92 on binary predicates, and the approach scales to 100K facts and to real knowledge graphs. Reliable decoding in turn unlocks capabilities that trained embeddings and exact reasoners do not combine: open-vocabulary reasoning over entities unseen at training, graceful robustness to noise, and multi-constraint queries. The framework trades exactness for efficiency: it suits approximate reasoning over large knowledge bases with graceful degradation, but not domains requiring formal guarantees. More broadly, the results suggest that effective reasoning in lossy vector representations comes from algorithms that exploit task structure, not from better encodings alone.

1 Introduction

A recurring goal in AI is to combine the flexibility of neural representations with the structure and interpretability of symbolic reasoning. Vector Symbolic Architectures (VSAs) are one of the few substrates where symbolic structure lives *natively* in vector space: high-dimensional vectors combined through binding and superposition form distributed representations that preserve compositional structure (Kanerva, 2009; 2010). The appeal is concrete. Facts become fixed-width vectors; inference reduces to a handful of constant-time algebraic operations that run in parallel on GPUs and map onto neuromorphic hardware; new predicates and entities are just fresh random vectors, so generalization is zero-shot; and answers carry similarity scores and degrade gracefully instead of failing outright. This profile makes VSAs a candidate for the structured, differentiable, hardware-efficient memory that neuro-symbolic systems and memory-augmented neural agents increasingly explore (Langenegger et al., 2023; Bronzini et al., 2025): a store a network could write to, compose, and query without abandoning gradients or GPUs. Yet they have delivered on this promise only for shallow tasks such as analogy-making (Gayler, 2003; Goldowsky & Sarathy, 2024) and cognitive modeling (Hersche et al., 2023); *relational* reasoning at scale (storing many facts and answering structured queries over them) has remained out of reach.

The obstacle is interference, and it is fundamental. Encoding a k-argument fact superimposes k bound terms in a single vector, so recovering any one argument leaves cross-talk from the other $k-1$; we prove this drives the signal-to-noise ratio down as $1/\sqrt{k}$ (Section 4): a floor for query-agnostic role-filler decoding, not an implementation artifact, that no choice of random vectors escapes. (Our method escapes it not with better vectors but by exploiting what the query already fixes; the floor binds any decoder that ignores that information.) This is why, for over a decade, VSAs have excelled at shallow associative tasks yet stalled on relational reasoning: the representation itself appears to forbid it. Prior work largely treated interference as

a problem of *representation* (seeking better encodings or factorization schemes that pack more recoverable structure into a vector (Plate, 2003; Frady et al., 2020; Langenegger et al., 2023)) and hit that floor.

Our starting point is that the failure is *algorithmic*, not representational. Standard decoding treats unbinding as a one-shot operation (Kleyko et al., 2023b;a) that ignores what the query already tells you. A query fixes most of a fact's structure; that known structure can be *subtracted away* before decoding rather than fought through a cleaner encoding. Effective reasoning in lossy vector representations, we argue, demands query-aware decoding: algorithms that exploit known constraints to actively cancel interference.[1]

Two things make this matter beyond VSA. Practically, it turns a stalled substrate into a usable one: we demonstrate reliable, scalable relational *retrieval* and single-rule reasoning over real knowledge graphs, and show that this reliability *unlocks* capabilities trained embeddings and exact reasoners do not combine: open-vocabulary reasoning over unseen entities, graceful noise robustness, and multi-constraint queries. This is the kind of compositional, training-free store a neuro-symbolic or memory-augmented neural system could build on. (Wiring that store into a downstream neural or agentic system is motivation we leave to future work.) Conceptually, the lesson is not tied to VSA: when a representation is lossy, decoding well means exploiting what the task already fixes, not merely engineering a cleaner encoding, a principle that transfers to other distributed and attention-based representations.

1.1 Contributions

We (i) prove a $1/\sqrt{k}$ SNR bound that explains role-filler binding's arity-dependent failure (Section 4); (ii) introduce a hybrid predicate-binding/positional-shift encoding that is both distinguishable and invertible (Section 5); (iii) develop successive interference cancellation (SIC), a query-aware decoding procedure with proven convergence (Section 6); (iv) validate it empirically, improving similarity from 0.31 to 0.92, scaling to 100K facts and to real knowledge graphs (Kinship, UMLS, Countries), and supporting multi-variable and multi-hop queries (Section 7); and (v) show that reliable decoding *unlocks* capabilities that trained embeddings and exact reasoners do not combine (open-vocabulary reasoning over entities unseen at training, graceful robustness to noise, and multi-constraint conjunctive queries; Section 7.7), while characterizing throughout the method's guarantees and limits (Section 9).

2 Background: Vector Symbolic Architectures

A Vector Symbolic Architecture (VSA) represents every symbol (an entity or a predicate) as a *hypervector*: a point in a d-dimensional space with d large (here $d = 8192$). We use the Fourier Holographic Reduced Representation (FHRR) model (Plate, 2003; Frady et al., 2020), in which each hypervector is a tuple of d unit-magnitude complex phasors and atomic symbols are drawn independently and uniformly at random. The similarity of two hypervectors is the normalized magnitude of their complex inner product, $\mathrm{sim}(\mathbf{a}, \mathbf{b}) = |\mathbf{a} \cdot \mathbf{b}|/(\|\mathbf{a}\|\|\mathbf{b}\|) \in [0, 1]$.

The representation rests on one statistical fact. In high dimension, two independently random hypervectors are *nearly orthogonal*: their expected similarity is 0 and fluctuations shrink as $1/\sqrt{d}$, a concentration-of-measure effect. Thousands of unrelated symbols can therefore coexist while remaining pairwise distinguishable, and a slightly corrupted hypervector still resembles its noise-free original far more than any other symbol. This is what makes approximate, distributed storage reliable.

VSAs provide three algebraic operations. Their *differing* guarantees, summarized below, are what our method exploits.

Bind ($\otimes$). Binding composes two hypervectors into a third that is dissimilar to both inputs. In FHRR it is elementwise complex multiplication, so it is commutative, associative, and distributes over addition. Binding is *approximately invertible*: an *unbind* operation (binding with the complex conjugate) recovers one operand from the other, $\mathrm{unbind}(\mathbf{a} \otimes \mathbf{b}, \mathbf{a}) \approx \mathbf{b}$. Recovery is not exact; each bind/unbind introduces a small residual

[1] This differs from attention-based "query-aware" decoding in neural models, which relies on learned attention weights. Our approach exploits known query constraints algebraically.

error ϵ, with $\|\epsilon\| \approx 0.05$ at $d = 8192$ and scaling as $1/\sqrt{d}$. Binding is how VSAs form *associations*, such as attaching a filler to a role or a value to a key. The signal-to-noise results in Sections 4–6 are expected-value statements over i.i.d. random hypervectors in this large-d regime.

Bundle ($\oplus$). Bundling superimposes hypervectors by (normalized) summation. Unlike binding, the result is *similar* to each constituent, so bundling implements an unordered set. It is commutative and associative but lossy and *not* invertible: as more vectors are superimposed, mutual crosstalk grows and individual terms become harder to recover. A bundle keeps its members retrievable only while their number is small relative to d.

Permute (π). Permutation applies a fixed reordering of coordinates; we use circular shift by n positions, π_n. It maps a hypervector to a dissimilar one, preserves inner products, and is *exactly* invertible with no approximation error: $\pi_{-n}(\pi_n(\mathbf{v})) = \mathbf{v}$. Permutation is the standard device for encoding order or position, and its exactness, in contrast to the lossy bind, is central to our method.

Cleanup. Algebraic operations yield noisy hypervectors; to read a discrete symbol back out we perform *cleanup* against a codebook of known atomic vectors, returning the nearest neighbor $\hat{s} = \arg\max_s \mathrm{sim}(\mathbf{q}, \mathbf{v}_s)$. Cleanup succeeds when the query's similarity to the true symbol exceeds its similarity to every distractor.

Together, these operations let a VSA build fixed-width records from atomic symbols and query them with constant-time algebra, trading exact recovery for a compact, noise-tolerant representation (Kanerva, 2019).

3 Problem Formulation

We consider relational knowledge bases built from ground atoms $p(t_1, \ldots, t_k)$, where p is a predicate from a finite set $\mathcal{P}$, each t_i is an entity from a finite set $\mathcal{E}$, and k is the predicate's arity. A knowledge base $\mathcal{K}$ is a finite set of such atoms, for instance the family relation {`parent(alice,bob)`, `parent(bob,carol)`, ...}. Every symbol $s \in \mathcal{P} \cup \mathcal{E}$ is assigned a random hypervector $\mathbf{v}_s \in \mathbb{C}^d$, and an encoding function $\phi : \mathcal{P} \times \mathcal{E}^k \to \mathbb{C}^d$ maps each atom to a single hypervector; the knowledge base is stored as the collection $\{\phi(a) : a \in \mathcal{K}\}$, one vector per fact. Because each symbol's vector is drawn independently of all others, the vocabulary need not be fixed in advance: a new entity or predicate is simply a fresh random vector that can be added at any time, with no retraining, a property we exploit for open-vocabulary reasoning (Section 7.7). We do assume queries are answerable (at least one fact matches); no-answer and negated queries are out of scope.

Queries and their semantics. A query is an atom with one argument replaced by a variable, such as `parent(alice,`X`)`, or more generally $p(t_1, \ldots, t_{i-1}, X, t_{i+1}, \ldots, t_k)$. Its intended answer is exactly the classical relational one,

$$[\![p(\ldots, X, \ldots)]\!]_{\mathcal{K}} = \{ t \in \mathcal{E} : p(t_1, \ldots, t, \ldots, t_k) \in \mathcal{K} \}.$$

The task is to recover this answer set *through vector operations and cleanup* rather than by symbolic table lookup. Because the representation is lossy, we accept *approximate* answers: each candidate is returned with a similarity score, and retrieval is deemed correct when cleanup ranks the ground-truth entity highest. This is the precise sense in which our reasoning is approximate: answers agree with the exact relational semantics up to similarity-ranked error, not by logical proof.

Encoding requirements. For this to work, ϕ must satisfy three properties. First, *predicate distinguishability*: $\mathrm{sim}(\phi(p, \mathbf{t}), \phi(p', \mathbf{t})) \approx 0$ when $p \neq p'$, so that `parent` and `enemy` facts are never confused. Second, *positional recoverability*: given $\mathbf{f} = \phi(p, t_1, \ldots, t_k)$ and a position i, the entity t_i can be recovered by cleanup. Third, *bounded interference*: the noise incurred when decoding one argument, or when superimposing several facts, must stay small enough for cleanup to succeed. Section 4 shows that the standard encoding fails property (2) as arity grows, and Sections 5–6 restore it.

4 Encoding Strategies

4.1 Role-Filler Binding

The standard approach encodes a fact as:

$$\phi_{\mathrm{RF}}(p, t_1, \ldots, t_k) = \mathbf{p} \otimes \left(\oplus_{i=1}^{k}(\rho_i \otimes \mathbf{t}_i)\right) \tag{1}$$

where ρ_i is a random role vector for position i. To decode position j, we unbind both the predicate and the role:

$$\mathbf{q}_j = \mathrm{unbind}\left(\mathrm{unbind}(\mathbf{f}, \mathbf{p}), \rho_j\right) \tag{2}$$

Expanding the unbind operations reveals the problem:

$$\mathrm{unbind}(\mathbf{f}, \mathbf{p}) \approx \oplus_{i=1}^{k}(\rho_i \otimes \mathbf{t}_i) \tag{3}$$

$$\begin{aligned} \mathbf{q}_j &\approx \mathrm{unbind}\left(\sum_{i=1}^{k}(\rho_i \otimes \mathbf{t}_i), \rho_j\right) \\ &= \mathbf{t}_j + \sum_{i \neq j} \mathrm{unbind}(\rho_i \otimes \mathbf{t}_i, \rho_j) \end{aligned} \tag{4}$$

The term $\mathrm{unbind}(\rho_i \otimes \mathbf{t}_i, \rho_j) = \mathbf{t}_i \otimes \rho_i \otimes \rho_j^{\dagger}$ is approximately a random vector since ρ_i and ρ_j are independent.

Proposition 1 (Role-Filler SNR Bound)**.** *For role-filler encoding with k arguments, the expected signal-to-noise ratio at position j is:*

$$SNR_{RF} = \frac{1}{k-1}$$

and the expected cosine similarity to the target entity is:

$$\mathbb{E}[sim(\mathbf{q}_j, \mathbf{t}_j)] \approx \frac{1}{\sqrt{k}}$$

for large d and random orthogonal vectors.

Proof. Decoding position j unbinds the role ρ_j from the argument bundle, which splits into a signal term and a sum of noise terms:

$$\mathbf{q}_j = \mathrm{unbind}\Big(\textstyle\sum_{i=1}^{k} \rho_i \otimes \mathbf{t}_i,\ \rho_j\Big) = \underbrace{\mathbf{t}_j}_{\text{signal}} + \underbrace{\sum_{i \neq j} \mathbf{n}_i}_{\text{noise}}, \qquad \mathbf{n}_i := \mathrm{unbind}(\rho_i \otimes \mathbf{t}_i, \rho_j).$$

The signal term appears because unbinding ρ_j from its own bound pair $\rho_j \otimes \mathbf{t}_j$ returns $\mathbf{t}_j$. Each noise term expands to $\mathbf{n}_i = \mathbf{t}_i \otimes \rho_i \otimes \rho_j^{\dagger}$, a binding of three mutually independent random vectors, and is therefore itself approximately a random unit vector, orthogonal in expectation to $\mathbf{t}_j$ and to every other $\mathbf{n}_{i'}$.

We now read off the SNR. The signal has unit power, $\|\mathbf{t}_j\|^2 = 1$. The noise is a sum of $k-1$ mutually orthogonal unit vectors, so cross terms vanish in expectation and

$$\mathbb{E}\Big[\big\|\textstyle\sum_{i \neq j} \mathbf{n}_i\big\|^2\Big] = \textstyle\sum_{i \neq j} \mathbb{E}[\|\mathbf{n}_i\|^2] = k-1,$$

giving $\mathrm{SNR} = 1/(k-1)$. For the cosine similarity, orthogonality of the noise to $\mathbf{t}_j$ gives $\mathbb{E}[\mathbf{q}_j \cdot \mathbf{t}_j] = 1$, while $\mathbb{E}[\|\mathbf{q}_j\|^2] = 1 + (k-1) = k$; hence $\mathbb{E}[\mathrm{sim}(\mathbf{q}_j, \mathbf{t}_j)] \approx 1/\sqrt{k}$. □

For binary predicates ($k = 2$), this predicts similarity around $1/\sqrt{2} \approx 0.707$. Ternary predicates drop to $1/\sqrt{3} \approx 0.577$. The $1/\sqrt{k}$ degradation appears fundamental to role-filler encoding, not fixable with better random vectors.

The empirical similarity is worse still. In the encoding-comparison experiment of Section 7 (Table 1; $d = 8192$, FHRR, family-tree `parent` facts), binary role-filler decoding attains only ≈ 0.31, well below the 0.707 the idealized SNR bound predicts. The shortfall comes from a source the idealized analysis ignores: accumulated approximation error in the bind/unbind chain. Each bind or unbind introduces a residual error $\epsilon_{\text{bind}} \approx 0.05$ (Section 2), and encoding then decoding a k-ary fact chains k binds and one unbind. A first-order estimate that discounts the idealized value by these operations,

$$\mathbb{E}[\text{sim}] \;\approx\; \frac{1}{\sqrt{2}}\,(1-\epsilon_{\text{bind}})^3 \;\approx\; 0.707 \times 0.86 \;\approx\; 0.61 \qquad (k=2),$$

already halves the bound; the remaining gap to 0.31 reflects further finite-d effects and normalization after bundling that this back-of-the-envelope model omits. We do not claim the estimate is tight: its purpose is to show that the observed collapse is quantitatively consistent with *compounding bind error*, which motivates an encoding that avoids that chain entirely (Sections 5–6).

4.2 Shift-Based Encoding

An alternative approach uses positional shifts instead of role binding:

$$\phi_{\text{Shift}}(p, t_1, \ldots, t_k) = \sum_{i=1}^{k} \pi_i(\mathbf{t}_i) \tag{5}$$

The predicate p is not encoded in the vector itself; instead, the knowledge base is partitioned by predicate, so predicate identity is implicit in which partition stores the fact. Decoding position j works by applying the inverse shift:

$$\mathbf{q}_j = \pi_{-j}\left(\sum_{i=1}^{k} \pi_i(\mathbf{t}_i)\right) = \mathbf{t}_j + \sum_{i \neq j} \pi_{i-j}(\mathbf{t}_i) \tag{6}$$

Proposition 2 (Shift-Based SNR Bound)**.** *Shift-based encoding with k arguments has the same expected positional signal-to-noise ratio as role-filler binding, SNR $= 1/(k-1)$, but incurs no bind/unbind approximation error, because both its encoding and its decoding use only exactly-invertible permutations.*

Proof. After decoding, $\mathbf{q}_j = \mathbf{t}_j + \sum_{i \neq j} \pi_{i-j}(\mathbf{t}_i)$. Each shifted term $\pi_{i-j}(\mathbf{t}_i)$ is orthogonal to $\mathbf{t}_j$ in expectation for random entities, so by the argument of Proposition 1 the signal-to-noise ratio is again $1/(k-1)$. The difference is in the error terms, not the SNR: permutation is exactly invertible ($\pi_{-n}(\pi_n(\mathbf{v})) = \mathbf{v}$), so unlike the role-filler chain there is no accumulated bind/unbind residual ϵ. □

Empirically this matters, because it is the accumulated ϵ, not the SNR, that drives the role-filler collapse: without that compounding error, shift-based decoding attains the SNR-predicted level ($\sim$0.71 for binary), whereas role-filler falls to $\sim$0.31 at the same dimension and data (Table 1).

The cost is predicate distinguishability. Shift-based encoding makes `parent(alice, bob)` and `enemy(alice, bob)` identical unless you store them in separate partitions.

4.3 Alternative Encoding Strategies

We also considered alternatives. Predicate-as-slot binding (the predicate as role 0) merely redistributes the same arity-dependent noise across a different layout, still suffering the $1/\sqrt{k}$ degradation of Proposition 1. Tensor or outer-product encodings grow exponentially with arity, breaking the constant-time retrieval that makes VSA appealing. Learned role vectors would sacrifice zero-shot reasoning and the orthogonality our SNR analysis relies on.

Pure shift-based encoding achieves clean positional decoding but loses predicate identity, requiring external indexing. The hybrid encoding we introduce next occupies a Pareto-optimal point in this design space: it preserves predicate distinguishability through binding, enables invertible positional decoding through shifts, and maintains bounded interference with provable SNR guarantees.

5 Hybrid Encoding with Predicate Binding

We propose a hybrid approach combining the strengths of both methods:

Definition 1 (Hybrid Encoding)**.**

$$\phi_{Hybrid}(p, t_1, \ldots, t_k) = \mathbf{p} \otimes \left(\sum_{i=1}^{k} \pi_i(\mathbf{t}_i) \right) \tag{7}$$

The design uses binding for the predicate (to distinguish different relations) and shifts for positions (to enable invertible decoding). To decode position j:

$$\mathbf{a} = \text{unbind}(\mathbf{f}, \mathbf{p}) \approx \sum_{i=1}^{k} \pi_i(\mathbf{t}_i) \tag{8}$$

$$\mathbf{q}_j = \pi_{-j}(\mathbf{a}) = \mathbf{t}_j + \sum_{i \neq j} \pi_{i-j}(\mathbf{t}_i) \tag{9}$$

Proposition 3 (Hybrid Encoding Properties)**.** *The hybrid encoding achieves:*

1. ***Predicate distinguishability****: $sim(\phi(p, \mathbf{t}), \phi(p', \mathbf{t})) \approx 0$ for $p \neq p'$*
2. ***Positional SNR****: Same as shift-based encoding (Proposition 2)*
3. ***Minimal overhead****: Bind/unbind error contributes $\mathcal{O}(\epsilon)$ noise where $\epsilon \approx 0.05$ for $d = 8192$*

Proof. Different predicates use independent vectors with $\mathbb{E}[\mathbf{p} \cdot \mathbf{p}'] = 0$. After unbinding, $\text{unbind}(\mathbf{f}, \mathbf{p}) \approx \sum_{i=1}^{k} \pi_i(\mathbf{t}_i) + \epsilon$ where $\|\epsilon\| = \mathcal{O}(0.05)$ for FHRR at $d = 8192$. Positional SNR follows Proposition 2. □

6 Successive Interference Cancellation

The key observation is that in query answering, we often have *known constraints* that can be exploited to remove interference before cleanup. We borrow both the name and the core idea from *successive interference cancellation* in multiuser detection (Verdú, 1998), where a receiver peels off the signals of already-decoded users to clean up the rest; here the "users" are a fact's arguments and their "signals" are the shifted contributions we subtract.

6.1 The Cancellation Algorithm

Algorithm 1 handles a query $p(t_1, \ldots, t_{i-1}, X, t_{i+1}, \ldots, t_k)$ in which every position except i is bound. It unbinds the predicate to recover the argument bundle, subtracts each known argument's shifted contribution to cancel its interference, and then applies the inverse shift and cleanup to the residual, so that, ideally, only the target argument remains to be decoded.

This resembles classical relevance-feedback methods such as the Rocchio algorithm (Rocchio, 1971), in which partial results refine the query vector, but here the refinement is *subtractive* (it cancels known structure rather than adding it) and operates within the vector space itself, below the symbolic layer, rather than at the retrieval level.

Proposition 4 (Cancellation Convergence)**.** *For a k-ary predicate with m bound arguments, interference cancellation reduces the noise power from $(k-1)$ to $(k-1-m)$. For a binary predicate with one bound argument, the decoded vector is:*

$$\mathbf{q}_i \approx \mathbf{t}_i + \mathcal{O}(\epsilon)$$

where ϵ is the bind/unbind approximation error.

Algorithm 1 Successive Interference Cancellation

Input: Fact vector $\mathbf{f}$, predicate $\mathbf{p}$, bound arguments $\{(j, t_j)\}_{j \neq i}$, query position i
Output: Retrieved entity $\hat{t}_i$

1: $\mathbf{a} \leftarrow \text{unbind}(\mathbf{f}, \mathbf{p})$ {Recover argument bundle}
2: **for** each bound position $j \neq i$ **do**
3: $\quad \mathbf{c}_j \leftarrow \pi_j(\mathbf{t}_j)$ {Compute contribution}
4: $\quad \mathbf{a} \leftarrow \mathbf{a} - \mathbf{c}_j$ {Cancel interference}
5: **end for**
6: $\mathbf{q}_i \leftarrow \pi_{-i}(\mathbf{a})$ {Decode cleaned bundle}
7: $\hat{t}_i \leftarrow \arg\max_{t \in \mathcal{E}} \text{sim}(\mathbf{q}_i, \mathbf{t})$ {Cleanup}
8: **return** $\hat{t}_i$

Proof. Unbinding the predicate recovers the argument bundle up to the bind/unbind residual:

$$\mathbf{a} = \text{unbind}(\mathbf{f}, \mathbf{p}) = \sum_{j=1}^{k} \pi_j(\mathbf{t}_j) + \epsilon_{\text{unbind}}, \qquad \|\epsilon_{\text{unbind}}\| = \mathcal{O}(\epsilon).$$

For each of the m bound arguments t_{j_ℓ} we form its exact shifted contribution $\pi_{j_\ell}(\mathbf{t}_{j_\ell})$ and subtract it. This is the crux: because the bundle is a plain linear sum, subtracting a known term removes it *exactly*, leaving

$$\mathbf{a}' = \mathbf{a} - \sum_{\ell=1}^{m} \pi_{j_\ell}(\mathbf{t}_{j_\ell}) = \sum_{j \notin \{j_1, \ldots, j_m\}} \pi_j(\mathbf{t}_j) + \epsilon_{\text{unbind}}.$$

The residual now contains only the $k - m$ still-unknown arguments, so decoding the target position i faces $k - m - 1$ interference terms rather than $k - 1$; the noise power falls from $(k - 1)$ to $(k - 1 - m)$. In the binary case ($k = 2$, $m = 1$) all interference is cancelled and only the target and the residual remain: $\mathbf{a}' = \pi_i(\mathbf{t}_i) + \epsilon_{\text{unbind}}$, so $\mathbf{q}_i = \pi_{-i}(\mathbf{a}') = \mathbf{t}_i + \pi_{-i}(\epsilon_{\text{unbind}})$. With $\|\epsilon_{\text{unbind}}\| \approx 0.05$ (FHRR, $d = 8192$), $\mathbb{E}[\text{sim}(\mathbf{q}_i, \mathbf{t}_i)] \approx 1/\sqrt{1 + \epsilon^2} \approx 0.999$. □

One subtlety: cancellation requires the argument bundle to be a plain *linear sum*, not a renormalized bundle, so that subtraction is exact.

6.2 A Worked Example (Our Running Thread)

We ground the method in one small world that the rest of the paper returns to. Three people form a two-fact family tree, `parent`(`alice`, `bob`) and `parent`(`bob`, `carol`), and every symbol (the predicate `parent` and the entities `alice`, `bob`, `carol`) is an independent random hypervector.

Encoding. The fact `parent`(`alice`, `bob`) becomes a single vector by shifting each argument to its position and binding the predicate over the sum,

$$\mathbf{f} = \mathbf{p} \otimes \big(\pi_1(\mathbf{alice}) + \pi_2(\mathbf{bob})\big).$$

The shift π_1 marks `alice` as the first argument and π_2 marks `bob` as the second, so position is recoverable; binding $\mathbf{p}$ stamps the relation, so `parent` facts never collide with, say, `sibling` facts over the same people.

Decoding the query `parent`(`alice`, ?). We know the predicate and the first argument and want the second (Figure 1). Unbinding $\mathbf{p}$ recovers the argument bundle $\pi_1(\mathbf{alice}) + \pi_2(\mathbf{bob})$, up to the $\approx 5\%$ bind residual. This is exactly where interference bites: to read out `bob` we invert π_2, but $\pi_1(\mathbf{alice})$ is still present as cross-talk. A naive one-shot decode leaves that term in place, and cleanup lands on `bob` at similarity only ≈ 0.71. Successive interference cancellation instead *subtracts* the contribution we already know, $\pi_1(\mathbf{alice})$ (exactly, because the bundle is a plain sum), leaving $\pi_2(\mathbf{bob})$ alone; applying π_{-2} and cleaning up recovers `bob` at ≈ 0.93. Had we used the standard role-filler encoding, this same query would have collapsed to ≈ 0.31 (Table 1): the interference from the known `alice` is precisely what SIC removes.

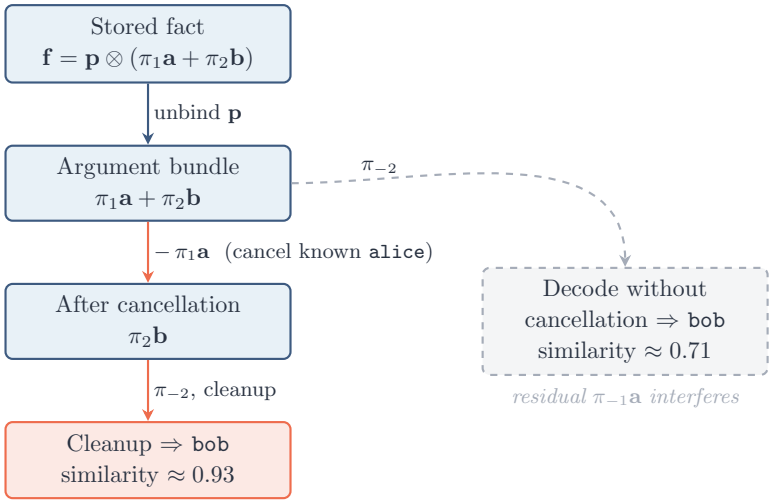

Figure 1: Successive interference cancellation on the query `parent(alice, ?)`. Subtracting the known argument's contribution before decoding removes interference, raising retrieval similarity from ≈ 0.71 to ≈ 0.93.

Why it recurs. This one example exercises the entire paper, and we flag each recurrence as it appears. Chaining `parent(alice, bob)` with `parent(bob, carol)` to derive `ancestor(alice, carol)` is the multi-hop test (Section 7.5); asking `parent(?, ?)` with *no* argument known is the hard multi-variable case; introducing a fourth person `dave` at query time, never seen during setup, is open-vocabulary reasoning (Section 7.7); and corrupting $\mathbf{f}$ before decoding is the robustness test.

6.3 Storage and Interference Scope

Facts can be stored either *separately* (each as its own vector, our default) or *bundled* into a single superposition $\mathbf{KB} = \sum_i \mathbf{f}_i$. Separate storage costs $O(n \cdot d)$ space but keeps every fact exactly retrievable and admits ANN indexing for sublinear query time. Bundling is $O(d)$ but capacity-limited (roughly $d/(k \log(1/\delta)) \approx 900$ facts for $d = 8192$, $k = 2$, $\delta = 0.01$), and its interference grows with n. We therefore use separate storage; the appendix gives the capacity derivation.

Crucially, our hybrid encoding applies superposition only *within* a fact, to combine its arguments ($\sum_i \pi_i(\mathbf{t}_i)$); the facts themselves stay independent. This intra-fact (rather than inter-fact) bundling is deliberate: a global superposition would make interference grow with knowledge-base size, whereas independent storage gives $O(1)$ decoding per fact, no cross-fact crosstalk, and natural parallelism. Predicate binding distinguishes structure *within* each fact; it does not merge facts together.

7 Experiments

We organize the evaluation as a progression from the core encoding result outward: we first establish that interference cancellation works and *why* (Section 7.2), then probe how it depends on dimension (Section 7.3) and knowledge-base size (Section 7.4); we then move from single-fact retrieval to reasoning (Section 7.5), from synthetic data to real knowledge graphs (Section 7.6), and finally to the capabilities that reliable decoding unlocks (Section 7.7). Each experiment closes with a short discussion, and Section 9 synthesizes across them.

7.1 Experimental Setup

What we manipulate and what we measure. Across the experiments the *independent* variables (what we deliberately vary) are the encoding method, the predicate arity k, the dimension d, the knowledge-base size n, the reasoning depth (number of chained hops), the number of arguments left unbound in a query, the additive-noise level η, and the fraction of entities withheld from a baseline. The core *dependent* question is

Table 1: Encoding comparison: mean similarity of the decoded argument to its true value, by method and arity. Empirical rows are mean ± standard deviation over 10 seeds (41–50) at $d = 8192$; theoretical rows are the analytical predictions from Section 4. The permutation-based encodings are essentially seed-invariant in the mean (±0.00), as their decoding uses only exact operations.

Encoding Method	**Binary**	**Ternary**	**Quaternary**
Role-filler	0.31±0.03	0.22±0.02	0.18±0.01
Shift-based	0.71±0.00	0.58±0.00	0.50±0.00
Hybrid (no cancel)	0.71±0.00	0.58±0.00	0.50±0.00
Hybrid + cancel	**0.92±0.00**	**0.70±0.00**	**0.50±0.00**
Theoretical (role-filler)	0.71	0.58	0.50
Theoretical (hybrid + cancel)	0.999	0.71	0.58

always the same: does cleanup recover the correct symbol, and how confidently? We report it through four metrics. *Retrieval similarity* is the normalized complex inner product $\mathrm{sim}(\mathbf{a}, \mathbf{b}) = |\mathbf{a} \cdot \mathbf{b}|/(\|\mathbf{a}\|\|\mathbf{b}\|) \in [0, 1]$ between the decoded vector and the true entity; a higher value is a stronger, more separable match and doubles as a confidence score. *valid@1* and *Hits@1* are the fractions of queries whose top-ranked answer forms a true fact, and is the specific held-out entity, respectively; we use valid@1 where a query has many correct answers and Hits@1 where it has one. *Recall/precision/F1* summarize set-valued queries, and we also report wall-clock query time. Unless stated otherwise we use the FHRR backend at $d = 8192$ with seed 42; symbol vectors are drawn independently from the unit sphere in $\mathbb{C}^{8192}$.

Datasets. The *synthetic* benchmarks isolate the effect of arity on otherwise-controlled data. A binary `parent` relation (99 facts over 20 people forming a 3-generation family tree) is our **running example** throughout; a ternary `teaches` relation (171 professor–course–semester triples) and a quaternary `transaction` relation (170 buyer–seller–item–price tuples) extend it to higher arity. Test queries mask one argument of a randomly chosen fact. The *real* benchmarks (Section 7.6) are three standard relational-reasoning knowledge graphs whose content is worth stating plainly: **Kinship** records kinship relations (e.g. `father`, `sister`) among the 104 members of the Alyawarra community; **UMLS** records 46 biomedical relations (e.g. `treats`, `causes`) among 135 medical concepts; and **Countries** records `locatedIn` and `neighborOf` among 271 countries, regions, and subregions, and is built so that answering requires *composing* these two relations: its S1/S2/S3 splits demand 1, 2, and 3 reasoning steps respectively. All symbols receive random hypervectors keyed by identifier; no embeddings are learned. Code, data, and seeds are released.

7.2 Encoding Comparison and Interference Cancellation

Our central claim is that role-filler binding fails on relational decoding for the reason Proposition 1 identifies, and that the hybrid encoding with successive interference cancellation removes that failure. We test this directly by comparing all four encodings (role-filler, shift-based, hybrid without cancellation, and hybrid with cancellation) across the three arities, reporting the mean similarity of the decoded argument to its true value.

Results. Table 1 and Figure 2 (left) show the predicted pattern. Role-filler decoding collapses with arity (0.31 / 0.22 / 0.18), consistent with the $1/\sqrt{k}$ trend of Proposition 1 once accumulated bind error is included (Section 4). Shift-based and uncancelled-hybrid encodings both recover the shift-based level (0.71 / 0.58 / 0.50), confirming that exact permutations avoid the bind-error penalty while predicate binding adds distinguishability at negligible cost. Cancellation then lifts binary similarity to 0.92. This is below the 0.999 ceiling of the idealized analysis (Proposition 4), and the $\approx 8\%$ gap is larger than the bind/unbind residual alone would explain ($\epsilon^2 \approx 0.0025$ accounts for well under 1%): the dominant contributions are finite-d fluctuations in cleanup and the normalization applied after bundling, neither of which the idealized bound models. The point is qualitative (cancellation recovers most of the ideal gain) rather than a tight numerical match.

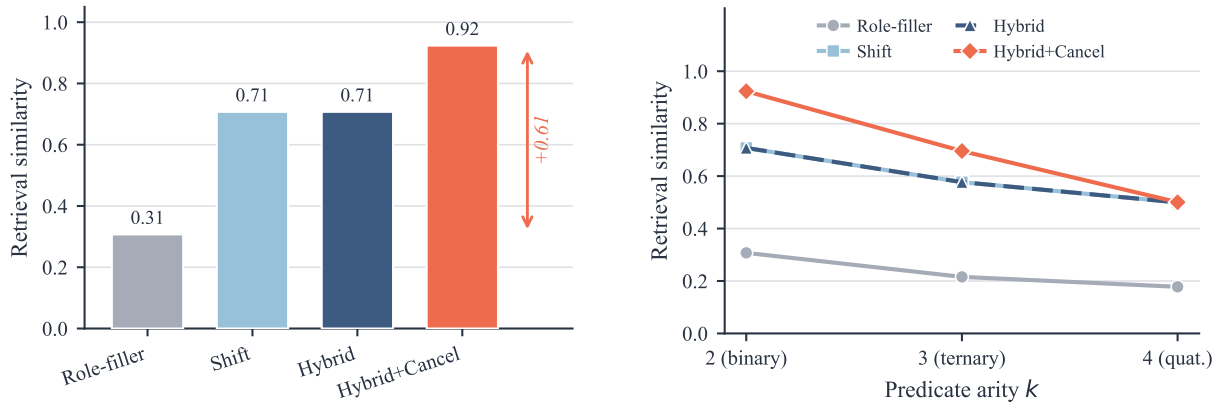

Figure 2: Left: encoding comparison on binary predicates, where cancellation reaches 0.92 versus 0.31 for role-filler. Right: similarity by arity, where cancellation's benefit shrinks because fewer of the $k-1$ interference terms can be removed with one bound argument. (Shift and hybrid-without-cancellation coincide; the latter is drawn dashed.)

Discussion. The gains are not uniform across arity, and their structure is itself informative. Cancellation effectiveness is proportional to $m/(k-1)$, the fraction of interference terms removed when m of the $k-1$ off-target arguments are known (Figure 2, right). For a binary predicate with one bound argument this fraction is 1 (every interference term is removed, explaining the jump from 0.31 to 0.92), whereas a ternary predicate with one bound argument removes only half ($0.22 \rightarrow 0.70$) and a quaternary with one bound argument removes a third and gains little (0.50). Cancellation thus helps most exactly where role-filler binding hurts most: low arity with most arguments known. This single relationship, $m/(k-1)$, will recur throughout: it explains the arity-dependent thresholds we adopt ($\theta \approx 0.85$ binary, 0.60 ternary), and it predicts the multi-hop and multi-variable behavior below, where the number of *known* constraints, not the raw arity, governs reliability.

7.3 Effect of Dimension

Dimension d is the one free hyperparameter of the encoding, so we ask what it actually buys. Sweeping d from 512 to 16384 across all three arities separates two effects that are easy to conflate. Retrieval accuracy is *not* what dimension controls: recovery stays 100% correct and the mean similarity is flat within each arity (binary ≈ 0.92, ternary ≈ 0.70, quaternary ≈ 0.50, matching Table 1) at every dimension tested. What dimension controls is *reliability*: the standard deviation of the similarity shrinks as $1/\sqrt{d}$, the same scaling as the bind/unbind error ϵ, so higher d yields tighter, more separable similarity distributions rather than higher means.

Discussion. This clarifies why $d = 8192$ is a reasonable default without being load-bearing: on these benchmark sizes even $d = 512$ decodes correctly, but the larger dimension provides margin (tighter distributions) and capacity headroom for larger vocabularies, at a cost that grows only linearly in d. It also empirically corroborates the error model of Section 2, since the observed $1/\sqrt{d}$ spread is exactly the predicted scaling of ϵ. The full per-arity sweep is in Appendix E.

7.4 Scaling to Large Knowledge Bases

Because facts are stored separately (Section 6), per-fact decoding is by construction independent of knowledge-base size; we verify this empirically by scaling from 100 to 100,000 facts. Table 2 confirms it: the top-ranked answer forms a valid fact on every query (valid@1 = 1.00) with mean similarity ≈ 0.93 across three orders of magnitude. (We report valid@1, whether the returned entity forms a true fact, rather

Table 2: Scaling behavior: retrieval quality is independent of knowledge-base size, while exhaustive-scan query time grows linearly. valid@1 is the fraction of queries whose top answer forms a true fact.

Facts	Query time (ms)	valid@1	Similarity
100	36	1.00	0.93
1,000	325	1.00	0.93
10,000	3,254	1.00	0.93
100,000	33,447	1.00	0.93

than exact match, because a query such as $\texttt{related}(a, X)$ admits many valid completions and exact match spuriously penalizes returning a different correct one.)

Discussion. The decoupling of quality from scale is the practical payoff of intra-fact (rather than inter-fact) superposition: each fact is decoded on its own, so adding facts adds candidates to search but never degrades the decode. The cost is linear query time under the exhaustive scan used here (tens of seconds at 100K), an implementation choice, not a property of the encoding: because retrieval reduces to nearest-neighbor search over independent vectors, approximate nearest-neighbor indexing (FAISS, HNSW) makes it sublinear while preserving the stable similarity. The scale at which cleanup itself would start to fail is set by capacity ($\sim d/(k \log(1/\delta))$; Appendix C), well beyond these sizes.

7.5 Reasoning: Chaining, Multi-Hop, and Multi-Variable Queries

We now move from decoding single stored facts to deriving new ones.

How chaining uses the VSA. It is worth making explicit what is symbolic here and what is vectorized, because it is easy to assume more (or less) is happening in the hypervectors than actually is. The *rules* are ordinary symbolic Horn clauses; we do *not* encode rules as hypervectors. What the VSA provides is the matching-and-storage substrate underneath them. To fire a rule such as $\texttt{ancestor}(X, Z) \leftarrow \texttt{parent}(X, Y), \texttt{ancestor}(Y, Z)$, the engine treats each body atom as a partially-bound query (e.g. $\texttt{parent}(\texttt{alice}, Y)$), uses SIC retrieval (unbind, cancel the known argument, cleanup) to recover the bindings that satisfy it, joins bindings across body atoms by their shared variables, re-encodes each satisfied head as a new fact vector through ϕ, and stores it (subject to a similarity-based novelty check that suppresses near-duplicate derivations). Unification is therefore approximate and vectorized, with two terms "match"-ing when their similarity clears a threshold θ, while the control flow (which rule to apply, in what variable order) is classical forward chaining. This is precisely the sense in which reasoning happens *through* vector operations: the logical scaffolding is symbolic, but every fact lookup, variable binding, and duplicate test is a VSA operation on hypervectors.

Forward chaining applies these rules to a fixpoint; we evaluate transitive closure on a 3-generation family tree (28 `parent` facts), from which the `ancestor` rule derives 68 facts, including our running $\texttt{ancestor}(\texttt{alice}, \texttt{carol})$, now *derived* through two VSA retrievals rather than stored. Recall degrades gracefully across hops ($0.96 \rightarrow 0.92 \rightarrow 0.69$ over 1–3 hops) while retrieval similarity stays high (0.92–0.93; Table 3), so error does not compound catastrophically with depth. Per-pair exact precision is low (0.07–0.23) for the metric reason noted above (an $\texttt{ancestor}(X, ?)$ query admits many valid descendants), so recall and similarity are the informative quantities here.

Queries with *multiple* unbound positions (e.g., `parent(?, ?)`) are harder, because cancellation needs at least one bound argument to work with. We handle them by decoding the first variable, cancelling its recovered value, and then decoding the second, returning the binding (e_1, e_2) with joint similarity $(s_1+s_2)/2$. On binary (`parent`) and ternary (`works_in(?, engineering, ?)`) datasets this reaches 100% precision and recall (Table 4) at joint similarity 0.78–0.81, below the single-variable level (0.92–0.93) because of the averaging, but well above typical retrieval thresholds.

Table 3: Multi-hop transitive closure: recall and similarity by hop depth. Recall degrades gracefully while similarity, the confidence of the answers that are retrieved, stays high.

Hops	Recall	Similarity
1	0.96	0.92
2	0.92	0.92
3	0.69	0.93

Table 4: Multi-variable query accuracy: retrieving all bindings for queries with multiple unbound positions.

Dataset	Precision	Recall	F1	Avg. Sim.
Binary (parent)	1.00	1.00	1.00	0.81
Ternary (works_in)	1.00	1.00	1.00	0.78

Finally, we implemented goal-directed backward chaining as an alternative control strategy: SLD resolution adapted to approximate VSA unification, recursing on rule bodies, unifying by similarity above threshold 0.5, with tabling to bound recursion. Across 14 test cases on recursive `ancestor` rules (ground facts, single-variable queries, 2-hop recursion via `ancestor(X,Z):-parent(X,Y),ancestor(Y,Z)`, and a pathological `loop(X):-loop(X)` that terminates through tabling), forward and backward chaining return the same answers with overlapping result sets, confirming semantic equivalence under approximate matching (Appendix D).

Discussion. Two things stand out. First, the graceful (rather than catastrophic) degradation with depth is exactly what the confidence-scored, separate-storage design predicts: each derived fact is decoded independently, so errors do not multiply, and the stable similarity means a threshold preserves precision as recall falls, a property exact reasoners, which return unranked all-or-nothing answers, cannot offer. Second, both the multi-hop drop at 3 hops and the multi-variable dip to 0.78–0.81 trace back to the same $m/(k-1)$ relationship from Section 7.2: reliability is governed by how many constraints are *known* at decode time, so deep chains and fully-unbound queries, where little is known, are the hard cases, precisely as the theory predicts. One subtlety underlies the multi-hop case in particular: the cancellation analysis (Proposition 4) assumes the known arguments are *exact* clean symbols, which holds for a ground query but only approximately in a chain, where the "known" arguments are themselves outputs of earlier decodes. This feedback of approximate arguments into later cancellations is part of why recall (not merely similarity) erodes with depth.

7.6 Real Knowledge Graphs

To test the method beyond synthetic benchmarks, we evaluate on three standard neuro-symbolic knowledge graphs (Das et al., 2018; Bouchard et al., 2015): Kinship (104 entities, 25 relations), UMLS (135 entities, 46 relations), and Countries (271 entities; designed specifically to require multi-hop rule reasoning), assigning a random hypervector per identifier. *Argument recovery* at this real vocabulary scale is perfect (Hits@1 = MRR = 1.00 on all three, at 5–13× the synthetic vocabulary), so the encoding and SIC decoding hold well beyond the toy setting. On Countries *multi-hop reasoning*, VSA-SIC with transitivity chaining solves the S1 task exactly (Hits@1 = 1.00), matching the perfect scores of ComplEx and NTP on that task (Trouillon et al., 2016; Rocktäschel & Riedel, 2017).

Discussion. The harder S2/S3 Countries variants require composing a second rule (a neighbor rule with the transitive closure), and here we hit an honest limit: the answers remain derivable (a symbolic reachability check recovers 23 of 24), but our forward-chaining engine does not reach them within a five-minute budget. The cause is specific and instructive: the reasoner performs each join as an $O(n)$ vector-similarity scan over stored facts, not an $O(1)$ indexed lookup, and novelty-checks each derived candidate against the growing knowledge base; a single-rule closure is cheap, but two-rule composition compounds this cost. The bottleneck

Table 5: Open-vocabulary / inductive reasoning: recovering arguments for facts about entities held out from a trained ComplEx model. VSA needs no training; the transductive baseline cannot score unseen entities without a full retrain.

Dataset	Novel/all	Facts	VSA Hits@1	ComplEx Hits@1
Kinship	31/104	14,135	**1.00**	0.00†
UMLS	40/135	5,867	**1.00**	0.00†

†Transductive: unseen entities are unscorable; a full retrain (66–137 s) is required to add them, versus VSA's $O(d)$ per-fact encode.

is the join, not the SIC decoding (which solves S1 and Protocol A exactly). This is a genuine tension in the "approximate joins by retrieval" premise: vector-retrieval joins avoid the enumeration blowup of large single-predicate retrieval but incur their own cost on deep multi-rule composition, where a classical indexed Datalog would not. Notably, S3 also collapses pure embedding models (ComplEx falls to 48% AUC-PR while rule learners such as NTP hold at 77% (Rocktäschel & Riedel, 2017)), so on S3 the limit is one of *performance* for our method but of *expressivity* for KGE. Appendix F gives the full protocol and baseline comparison.

7.7 What Reliable Decoding Unlocks

Removing the interference wall is not merely a better number: it makes usable a set of capabilities the wall previously blocked, and that exact reasoners or trained embeddings do not jointly offer. We demonstrate three (full protocols and tables in Appendix G).

Open-vocabulary, training-free reasoning. Because entities are random vectors and decoding needs no training, facts about entities introduced *at query time* are handled immediately. We test this as a *fact-recovery* task, as if our `dave` appeared only at query time: hold out 30% of entities from a trained ComplEx model, then, given a stored fact that involves a held-out entity with one argument masked, recover the missing argument. VSA-SIC does so at Hits@1 = 1.00 (∼10–16 ms/fact, no training), whereas transductive ComplEx cannot even score a fact containing an unseen entity (it has no embedding for one), so it scores 0 *by construction* and must fully retrain (66–137 s) to incorporate them (Table 5). The point is not that VSA is a better predictor, but that it handles an open vocabulary at all, training-free. (Inductive link-*prediction* methods such as GraIL and NBFNet (Teru et al., 2020; Zhu et al., 2021) also generalize to unseen entities, but by *learning* to predict new links from graph structure, a different task from the training-free recovery of stored facts measured here, and one we do not attempt.)

Robustness with calibrated confidence. Corrupting each stored fact vector with additive noise of relative magnitude η (our $\mathbf{f} = \phi(\texttt{parent}(\texttt{alice}, \texttt{bob}))$ included), recovery stays perfect up to $\eta = 2$ (noise twice the signal), then degrades *gracefully* to chance by $\eta = 16$, with the retrieval similarity tracking the corruption as a confidence signal (Figure 3). An exact or hash-indexed store fails on *any* corruption.

Multi-constraint conjunctive queries. SIC's per-argument cancellation makes reliable the conjunctive queries the interference wall previously collapsed to baseline: seeking entities that satisfy $r_1(X, a) \wedge r_2(X, b)$ by retrieving each constraint's candidate set and intersecting reaches precision = recall = F_1 = 1.00 on Kinship and UMLS.

Discussion. These are not three unrelated tricks but three faces of one property: an approximate relational store that is *open-vocabulary* (new symbols cost one random vector), *robust* (graceful, confidence-scored degradation), and *compositional* (multi-constraint queries), all training-free. A trained embedding gives none of the first two and buys the third only within a fixed vocabulary; an exact symbolic store gives compositionality but is brittle to noise and offers no ranking. The contribution of reliable decoding is to make this combination usable, which is what turns VSA from a promising-but-unreliable substrate into one a neuro-symbolic or memory-augmented system could actually build on.

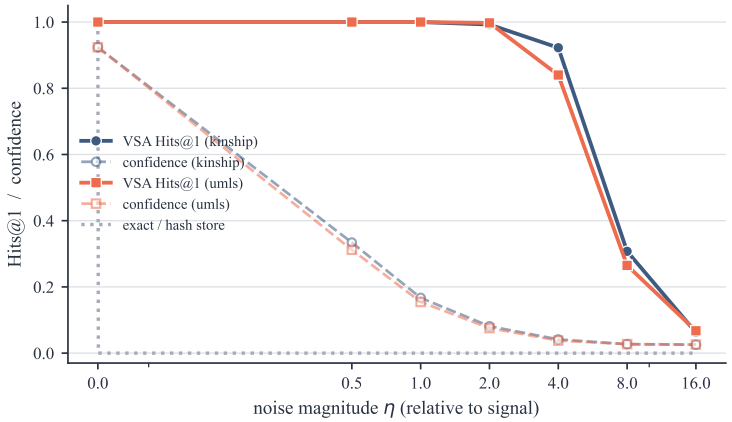

Figure 3: Graceful degradation under noise. VSA argument recovery (solid) stays perfect until the noise exceeds twice the signal, then declines smoothly to chance; the retrieval similarity (dashed) tracks the corruption as calibrated confidence. An exact or hash-indexed store (dotted) fails at any corruption.

8 Related Work

Classical logic engines (Prolog (Lloyd, 1987), ASP (Gelfond & Lifschitz, 1988)) provide exact inference with soundness and completeness guarantees, but query complexity grows exponentially with rule depth and fact count. Answer set solvers scale poorly beyond thousands of rules due to grounding bottlenecks. Probabilistic approaches (MLNs (Getoor & Taskar, 2007; De Raedt et al., 2016), ProbLog (De Raedt et al., 2007)) handle uncertainty but require expensive MCMC or lifted inference. Inductive logic programming systems (FOIL (Quinlan, 1990)) learn rules from data but lack real-time query capabilities.

Recent neuro-symbolic systems (Manhaeve et al., 2018; Evans & Grefenstette, 2018) embed logic in continuous spaces yet face opacity problems. Learned representations lack interpretability, and error propagation through deep networks makes debugging difficult. Graph neural networks can learn relational patterns but require large training sets and struggle with zero-shot generalization to unseen predicates. On knowledge-graph reasoning specifically, embedding models (ComplEx (Trouillon et al., 2016)) and differentiable rule learners (Neural LP (Yang et al., 2017), NTP (Rocktäschel & Riedel, 2017)) are the standard baselines, which we compare against on Kinship, UMLS, and Countries. A distinction we return to is that standard KG embeddings are *transductive*: they learn a vector per entity and cannot score an entity unseen at training. This motivated a line of *inductive* methods that generalize to new entities through graph structure, notably GraIL's subgraph reasoning (Teru et al., 2020) and NBFNet's path-based message passing (Zhu et al., 2021). Our method is inductive by a simpler route: because a new entity is just a fresh random vector, it needs no learning and no structural assumptions (Section 7.7), trading the ability to learn distributional regularities for immediate, training-free handling of an open vocabulary.

Classical VSA work (Plate, 2003; Kanerva, 2009) focused on encoding trees and sequences. HRR pioneered circular convolution for binding but provided no formal SNR analysis for multi-argument predicates. HDC introduced positional shifts for sequences but didn't address predicate distinguishability. Prior VSA applications (analogy, semantic parsing) used fixed templates; none attempted full logical reasoning with chaining. The closest decoding technique is the resonator network (Frady et al., 2020), which recovers the unknown factors of a single bound product by alternating cleanup across codebooks, and which in-memory factorizers scale by orders of magnitude on neuromorphic hardware (Langenegger et al., 2023). Our successive interference cancellation differs in both aim and mechanism: it does not search for unknown factors but exploits the *known* arguments of a query to subtractively remove their interference before a single cleanup, replacing iterative factor search with a direct algebraic cancellation.

Recent VSA systems approach relational reasoning by *learning*: LARS-VSA binds object features to symbolic rule vectors to curb interference (Mejri et al., 2024a), RESOLVE mixes relational and object-level

features through high-dimensional attention (Mejri et al., 2024b), and self-attention has been proposed as an alternative to resonator-style decoding (Yeung et al., 2024). These target learned relational abstraction; our contribution is analytic and training-free: an SNR characterization of arity-dependent interference paired with query-conditioned subtractive decoding. VSA is also increasingly used to interface with and probe neural representations (Bronzini et al., 2025), with open tooling lowering the barrier to entry (Heddes et al., 2023). To our knowledge this specific combination (query-aware interference cancellation with formal SNR bounds, validated on multi-hop retrieval) has not been reported.

9 Discussion

Taken together, the experiments trace a single thread. The encoding comparison (Section 7.2) establishes both the failure (role-filler decoding collapses as $1/\sqrt{k}$) and the fix, with the fraction $m/(k-1)$ governing exactly how much cancellation helps. That same quantity then predicts every subsequent result: dimension controls reliability but not accuracy (Section 7.3); quality is decoupled from knowledge-base size because facts decode independently (Section 7.4); reasoning degrades gracefully with hop depth and with the number of unbound variables, because both reduce the count m of known constraints (Section 7.5); the encoding holds at real vocabulary scale, and single-rule reasoning matches specialized baselines while multi-rule composition exposes the join (not the decoder) as the bottleneck (Section 7.6); and reliable decoding unlocks open-vocabulary, robust, compositional queries that no single alternative provides (Section 7.7). The method's strengths and its limits thus share one origin: it reasons by exploiting what the query already fixes, so it excels when much is known and degrades predictably when little is.

Why cancellation works. Three properties align. Linear superposition subtracts exactly ($\mathbf{a} = \mathbf{x} + \mathbf{y} \Rightarrow \mathbf{a} - \mathbf{x} = \mathbf{y}$); permutation is perfectly invertible (unlike bind/unbind, which loses $\approx 5\%$ each time); and the query tells us *what* to subtract. The third property is what distinguishes SIC from a better encoding: it exploits information that only exists at query time. It also bounds applicability: cancellation helps only when arguments are known, so fully-unbound queries fall back to baseline.

Guarantees. The approach offers two. Retrieval error is bounded: Proposition 4 cuts noise power from $(k-1)$ to $(k-1-m)$, leaving $\mathcal{O}(\epsilon)$. And every answer carries a similarity score (a ranking signal exact reasoners do not provide), so quality degrades gracefully (0.92/0.70/0.50 across arities) rather than abruptly, and a threshold trades recall for precision in a controlled way.

Approximate matching, not entailment. What we call unification is a similarity test: two structures unify when their similarity exceeds a threshold θ, a graded family whose strict limit ($\theta \to 1$) recovers exact matching and whose operating point is a design choice tied to arity and task. Because approximate steps can each clear θ while the composed inference drifts, we treat per-answer similarity as a confidence signal, never a claim of logical entailment.

9.1 Limitations

Several limits follow directly from the design; we state them plainly.

No logical guarantees. The method does not guarantee entailment: two facts at similarity 0.85 need not be logically equivalent, and chaining approximate steps compounds error, so multi-hop recall falls with depth ($0.96 \to 0.69$ over three hops). Consistency under negation is not assured: nothing prevents both `parent(alice, bob)` and its negation from scoring high once noise accumulates. The approach is therefore unsuitable where formal correctness is required.

Multi-rule chaining does not scale. Reasoning is reliable for shallow, single-rule composition (Countries S1), but two-rule composition (S2/S3) exposes the vector-similarity join, whose $O(n)$ per-step cost compounds across rules and fixpoint iterations where a classical indexed join would not. Two complementary directions could address this. Approximate nearest-neighbor indexing would cut the per-join constant. More promisingly, *query-directed* search (goal-driven strategies such as magic-sets rewriting or Monte Carlo tree search

that expand only query-relevant derivations rather than computing the full closure) targets the actual bottleneck: the S2/S3 answers are shallow (derivable in a few steps), so the cost is the *breadth* of the exhaustive fixpoint, not the depth of any single proof. Directing search toward the goal instead of enumerating all consequences is therefore the natural next step, and combines cleanly with the confidence scores SIC already provides as a search heuristic. Neither is implemented here; scaling multi-rule chaining is open.

Random-vector assumption. The SNR analysis relies on i.i.d. random symbol vectors; correlated, learned entity embeddings would break the orthogonality it assumes. Keeping symbols random buys training-free, zero-shot behavior at the cost of not exploiting distributional regularities in the data.

No learned generalization. Because the method stores and decodes rather than learns, it does not perform learned link prediction: on standard held-out KG completion it recovers only rule-derivable facts. Its niche is reliable, training-free retrieval and rule-based inference, complementary to trained predictors rather than a replacement for them.

Memory integration is motivation, not result. We demonstrate the store's properties (open-vocabulary, robust, compositional), but do not integrate it into a downstream neural or agentic system; that remains future work.

10 Conclusion

Query-aware interference cancellation makes VSA practical for scalable, approximate relational *retrieval*. By exploiting known query constraints to subtractively remove interference before cleanup, rather than seeking a better encoding alone, it lifts retrieval similarity from 0.31 to 0.92, holds quality stable to 100K facts and to real knowledge graphs, and supports single-rule multi-hop reasoning (deeper multi-rule composition remains bounded by chaining cost). Crucially, reliable decoding unlocks open-vocabulary, noise-robust, multi-constraint reasoning that trained embeddings and exact reasoners do not combine. The result is a similarity-ranked, bounded-error, constant-time complement to classical logic that positions VSA between brittle symbolic systems and opaque neural models. Future work includes scalable multi-rule chaining, tighter bounds for multi-variable decoding, and integrating this memory into neuro-symbolic and agentic systems.

Reproducibility Statement

All experiments use fixed random seeds (42 for the main runs; seeds 41–50 for the error bars in Table 1) and run on commodity CPU hardware, with no GPU required. We provide the complete code, datasets, and scripts to reproduce every table and figure as anonymized supplementary material with this submission; the two supporting libraries are bundled as source, so reproduction installs no author-identifying package. The release includes the synthetic-data generators, the real knowledge-graph splits (from the standard MINERVA distribution), the encoding and cancellation implementations, and the baseline configurations. Reported values are deterministic up to floating-point differences (± 0.03). Appendix H documents the full software stack, datasets, and hyperparameters, and Appendix A gives complete proofs of all propositions.

Broader Impact Statement

This is foundational work on synthetic and public-benchmark data with no human subjects; we foresee no direct harms. Because the method returns approximate answers without soundness guarantees, it should not be relied upon where formal correctness is required (e.g., safety-critical decisions).

References

Guillaume Bouchard, Sameer Singh, and Théo Trouillon. On approximate reasoning capabilities of low-rank vector spaces. In *AAAI Spring Symposium on Knowledge Representation and Reasoning*, 2015.

Marco Bronzini, Carlo Nicolini, Bruno Lepri, Jacopo Staiano, and Andrea Passerini. Hyperdimensional probe: Decoding LLM representations via vector symbolic architectures, 2025.

Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alex Smola, and Andrew McCallum. Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2018.

Luc De Raedt, Angelika Kimmig, and Hannu Toivonen. Problog: A probabilistic prolog and its application in link discovery. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2462–2467, 2007.

Luc De Raedt, Kristian Kersting, Sriraam Natarajan, and David Poole. Statistical relational artificial intelligence: Logic, probability, and computation. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 10(2):1–189, 2016.

Richard Evans and Edward Grefenstette. Learning explanatory rules from noisy data. *Journal of Artificial Intelligence Research*, 61:1–64, 2018.

E. Paxon Frady, Spencer Kent, Bruno A. Olshausen, and Friedrich T. Sommer. Resonator networks for factoring distributed representations of data structures, July 2020.

Ross W Gayler. Vector symbolic architectures answer jackendoff's challenges for cognitive neuroscience. *arXiv preprint cs/0412059*, 2003.

Michael Gelfond and Vladimir Lifschitz. The stable model semantics for logic programming. *ICLP/SLP*, 88:1070–1080, 1988.

Lise Getoor and Ben Taskar. *Introduction to Statistical Relational Learning*. MIT Press, Cambridge, MA, 2007.

Howard Goldowsky and Vasanth Sarathy. Analogical reasoning within a conceptual hyperspace. In *IJCAI Workshop on Analogical Abstraction in Cognition, Perception, and Language*, 2024.

Mike Heddes, Igor Nunes, Pere Vergés, Denis Kleyko, Danny Abraham, Tony Givargis, Alexandru Nicolau, and Alexander Veidenbaum. Torchhd: An open source python library to support research on hyperdimensional computing and vector symbolic architectures, 2023.

Michael Hersche, Mustafa Zeqiri, Luca Benini, Abu Sebastian, and Abbas Rahimi. A neuro-vector-symbolic architecture for solving Raven's progressive matrices. *Nature Machine Intelligence*, 5(4):363–375, April 2023. ISSN 2522-5839. doi: 10.1038/s42256-023-00630-8.

Pentti Kanerva. Hyperdimensional Computing: An Introduction to Computing in Distributed Representation with High-Dimensional Random Vectors. *Cognitive Computation*, 1(2):139–159, June 2009. ISSN 1866-9964. doi: 10.1007/s12559-009-9009-8.

Pentti Kanerva. What we mean when we say" what's the dollar of mexico?": Prototypes and mapping in concept space. In *AAAI fall symposium: quantum informatics for cognitive, social, and semantic processes*, pp. 2–6, 2010.

Pentti Kanerva. Computing with High-Dimensional Vectors. *IEEE Design & Test*, 36(3):7–14, June 2019. ISSN 2168-2356, 2168-2364. doi: 10.1109/MDAT.2018.2890221.

Denis Kleyko, Dmitri Rachkovskij, Evgeny Osipov, and Abbas Rahimi. A Survey on Hyperdimensional Computing aka Vector Symbolic Architectures, Part II: Applications, Cognitive Models, and Challenges. *ACM Computing Surveys*, 55(9):175:1–175:52, January 2023a. ISSN 0360-0300. doi: 10.1145/3558000.

Denis Kleyko, Dmitri A. Rachkovskij, Evgeny Osipov, and Abbas Rahimi. A Survey on Hyperdimensional Computing aka Vector Symbolic Architectures, Part I: Models and Data Transformations. *ACM Computing Surveys*, 55(6):1–40, July 2023b. ISSN 0360-0300, 1557-7341. doi: 10.1145/3538531.

Jovin Langenegger, Geethan Karunaratne, Michael Hersche, Luca Benini, Abu Sebastian, and Abbas Rahimi. In-memory factorization of holographic perceptual representations. *Nature Nanotechnology*, 18:479–485, 2023.

John W Lloyd. *Foundations of logic programming*. Springer Science & Business Media, 1987.

Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. DeepProbLog: Neural Probabilistic Logic Programming. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Mohamed Mejri, Chandramouli Amarnath, and Abhijit Chatterjee. LARS-VSA: A vector symbolic architecture for learning with abstract rules, 2024a.

Mohamed Mejri, Chandramouli Amarnath, and Abhijit Chatterjee. RESOLVE: Relational reasoning with symbolic and object-level features using vector symbolic processing, 2024b.

Tony A Plate. Holographic reduced representation: Distributed representation for cognitive structures. *CSLI Publications*, 2003.

J. Ross Quinlan. Learning logical definitions from relations. *Machine Learning*, 5(3):239–266, 1990.

J. J. Rocchio. Relevance feedback in information retrieval. In Gerard Salton (ed.), *The SMART Retrieval System: Experiments in Automatic Document Processing*, pp. 313–323. Prentice-Hall, 1971.

Tim Rocktäschel and Sebastian Riedel. End-to-end differentiable proving. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Komal K. Teru, Etienne Denis, and William L. Hamilton. Inductive relation prediction by subgraph reasoning. In *International Conference on Machine Learning (ICML)*, 2020.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International Conference on Machine Learning (ICML)*, 2016.

Sergio Verdú. *Multiuser Detection*. Cambridge University Press, 1998.

Fan Yang, Zhilin Yang, and William W. Cohen. Differentiable learning of logical rules for knowledge base reasoning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Calvin Yeung, Prathyush Poduval, and Mohsen Imani. Self-attention based semantic decomposition in vector symbolic architectures, 2024.

Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal Xhonneux, and Jian Tang. Neural Bellman-Ford networks: A general graph neural network framework for link prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

A Detailed Proofs

A.1 Proof of Proposition 1 (Role-Filler SNR Bound)

Proof. Consider the role-filler encoding of a k-ary predicate, $\mathbf{f} = \sum_{i=1}^{k}(\rho_i \otimes \mathbf{t}_i)$. To decode argument j we unbind with role vector ρ_j:

$$\mathbf{q}_j = \text{unbind}(\mathbf{f}, \rho_j) = \sum_{i=1}^{k} \text{unbind}(\rho_i \otimes \mathbf{t}_i, \rho_j).$$

When $i = j$ the unbind approximately recovers the signal, $\text{unbind}(\rho_j \otimes \mathbf{t}_j, \rho_j) \approx \mathbf{t}_j$. When $i \neq j$ we obtain noise terms $\mathbf{n}_i = \text{unbind}(\rho_i \otimes \mathbf{t}_i, \rho_j)$; for independent role vectors $\rho_i \perp \rho_j$ and random entities each $\mathbf{n}_i$ is approximately random and orthogonal to $\mathbf{t}_j$. Hence $\mathbf{q}_j = \mathbf{t}_j + \sum_{i \neq j} \mathbf{n}_i$.

Signal power. $\|\mathbf{t}_j\|^2 = 1$.

Noise power. Because the $\mathbf{n}_i$ are approximately orthogonal to each other,

$$\mathbb{E}\Big[\|\textstyle\sum_{i\neq j}\mathbf{n}_i\|^2\Big] = \sum_{i\neq j}\mathbb{E}[\mathbf{n}_i\cdot\mathbf{n}_i] + \sum_{i\neq i'}\mathbb{E}[\mathbf{n}_i\cdot\mathbf{n}_{i'}] \approx (k-1)+0.$$

Thus SNR $= 1/(k-1)$.

Cosine similarity. We have $\mathbb{E}[\|\mathbf{q}_j\|^2] = 1+(k-1) = k$ and $\mathbb{E}[\mathbf{q}_j\cdot\mathbf{t}_j] = 1+\sum_{i\neq j}\mathbb{E}[\mathbf{n}_i\cdot\mathbf{t}_j] \approx 1$, so

$$\mathbb{E}[\mathrm{sim}(\mathbf{q}_j,\mathbf{t}_j)] \approx \frac{1}{\sqrt{k}}.$$ □

A.2 Proof of Proposition 2 (Shift-Based SNR Bound)

Proof. For $\mathbf{f} = \sum_{i=1}^k \pi_i(\mathbf{t}_i)$, decoding position j applies the inverse shift:

$$\mathbf{q}_j = \pi_{-j}(\mathbf{f}) = \sum_{i=1}^k \pi_{i-j}(\mathbf{t}_i) = \mathbf{t}_j + \sum_{i\neq j}\pi_{i-j}(\mathbf{t}_i),$$

since $\pi_0(\mathbf{t}_j) = \mathbf{t}_j$ exactly. For random entities $\mathbb{E}[\pi_{i-j}(\mathbf{t}_i)\cdot\mathbf{t}_j] \approx 0$, so signal power is 1 and noise power is $k-1$, giving the same SNR $= 1/(k-1)$ as role-filler binding.

The crucial difference is exactness. Shifts satisfy $\pi_{-j}(\pi_j(\mathbf{v})) = \mathbf{v}$ with no approximation error, whereas bind/unbind introduces $\mathrm{unbind}(\rho\otimes\mathbf{v},\rho) = \mathbf{v}+\epsilon$ with $\|\epsilon\| \approx 0.05$ for FHRR at $d = 8192$. Role-filler encoding accumulates this error over its k binds and one unbind, while shift-based encoding accumulates none. □

Empirically this is the whole story: shift-based decoding attains $\sim 0.71 \approx 1/\sqrt{2}$, matching the SNR prediction, while role-filler falls to ~ 0.31 (Table 1): the gap is the accumulated ϵ, not a difference in SNR.

A.3 Proof of Proposition 3 (Hybrid Encoding Properties)

Proof. Write $\mathbf{a} = \sum_{i=1}^k \pi_i(\mathbf{t}_i)$ so that $\mathbf{f} = \mathbf{p}\otimes\mathbf{a}$.

Predicate distinguishability. For $p \neq p'$ with the same arguments, $\mathbf{f} = \mathbf{p}\otimes\mathbf{a}$ and $\mathbf{f}' = \mathbf{p}'\otimes\mathbf{a}$. For independent predicate vectors ($\mathbb{E}[\mathbf{p}\cdot\mathbf{p}'] = 0$), binding by different orthogonal vectors yields $\mathrm{sim}(\mathbf{f},\mathbf{f}') \approx \mathbb{E}[\mathbf{p}\cdot\mathbf{p}'] = 0$ up to error ϵ.

Positional SNR. Unbinding the predicate gives $\mathbf{a} = \mathrm{unbind}(\mathbf{f},\mathbf{p}) \approx \sum_i \pi_i(\mathbf{t}_i) + \epsilon_{\mathrm{unbind}}$, and $\pi_{-j}(\mathbf{a}) = \mathbf{t}_j + \sum_{i\neq j}\pi_{i-j}(\mathbf{t}_i) + \pi_{-j}(\epsilon_{\mathrm{unbind}})$. The positional interference contributes noise power $k-1$, matching shift-based encoding.

Minimal overhead. The extra term satisfies $\mathbb{E}[\|\pi_{-j}(\epsilon_{\mathrm{unbind}})\|^2] \approx \epsilon^2 \approx 0.0025$, so total noise power is $(k-1)+\mathcal{O}(\epsilon^2) \approx k-1$. The hybrid encoding thus preserves shift-based SNR while adding predicate distinguishability. □

A.4 Proof of Proposition 4 (Cancellation Convergence)

Proof. After unbinding the predicate, $\mathbf{a} = \sum_{j=1}^k \pi_j(\mathbf{t}_j) + \epsilon_{\mathrm{unbind}}$. Given m bound arguments at positions $\{j_1,\ldots,j_m\}$ and query position i, canceling each known contribution yields

$$\mathbf{a}' = \mathbf{a} - \sum_{\ell=1}^m \pi_{j_\ell}(\mathbf{t}_{j_\ell}) = \sum_{j\notin\{j_1,\ldots,j_m\}} \pi_j(\mathbf{t}_j) + \epsilon_{\mathrm{unbind}}.$$

Decoding position i,

$$\mathbf{q}_i = \pi_{-i}(\mathbf{a}') = \mathbf{t}_i + \sum_{\substack{j\notin\{j_1,\ldots,j_m\}\\ j\neq i}} \pi_{j-i}(\mathbf{t}_j) + \pi_{-i}(\epsilon_{\mathrm{unbind}}).$$

The residual interference has $(k-m-1)$ terms, giving noise power $(k-m-1)+\mathcal{O}(\epsilon^2)$; cancellation therefore reduces noise power from $(k-1)$ to $(k-1-m)$.

For the binary case $k=2$, $m=1$: $\mathbf{q}_i = \mathbf{t}_i + \pi_{-i}(\epsilon_{\text{unbind}})$, so

$$\mathbb{E}[\text{sim}(\mathbf{q}_i, \mathbf{t}_i)] \approx \frac{1}{\sqrt{1+\epsilon^2}} \approx 1 - \tfrac{\epsilon^2}{2} \approx 0.9987. \qquad \square$$

B Additional VSA Background

FHRR and FFT-based binding. FHRR implements binding as circular convolution, $(\mathbf{a} \otimes \mathbf{b})[i] = \sum_j a[j]\, b[(i-j) \bmod d]$, computable in $O(d \log d)$ via the FFT: $\mathbf{a} \otimes \mathbf{b} = \text{FFT}^{-1}(\text{FFT}(\mathbf{a}) \odot \text{FFT}(\mathbf{b}))$, with unbinding by element-wise division in the Fourier domain.

Orthogonality. For random unit vectors in $\mathbb{R}^d$ (or unit phasors in $\mathbb{C}^d$), $\mathbb{E}[\mathbf{a} \cdot \mathbf{b}] \approx 0$ and $\text{Var}[\mathbf{a} \cdot \mathbf{b}] \approx 1/d$. By a Johnson–Lindenstrauss argument, the probability that any of n random vectors collides (similarity > 0.9) is bounded by $\approx n^2/d$; at $d = 8192$, $n = 100$ this is below 0.12%. Binding preserves orthogonality in expectation: for $\mathbf{a} \perp \mathbf{b}$ and random $\mathbf{c}, \mathbf{d}$, $\mathbb{E}[(\mathbf{a} \otimes \mathbf{c}) \cdot (\mathbf{b} \otimes \mathbf{d})] \approx 0$.

Shifts. Circular shift $\pi_n(\mathbf{v})[i] = \mathbf{v}[(i-n) \bmod d]$ is exactly invertible ($\pi_{-n}\pi_n = \text{id}$), norm-preserving ($\|\pi_n(\mathbf{v})\| = \|\mathbf{v}\|$), and approximately orthogonality-preserving ($\mathbb{E}[\pi_n(\mathbf{v}) \cdot \mathbf{w}] \approx 0$ for $n \neq 0$). This combination is what makes shifts attractive for positional encoding.

C Full Capacity Derivation

With separate storage each fact is its own vector; capacity is bounded only by memory ($O(n \cdot d)$ for n facts) and every fact is exactly retrievable. Bundled storage superimposes facts, $\mathbf{KB} = \sum_{i=1}^{n} \mathbf{f}_i$, and cleanup succeeds when the target fact's similarity exceeds that of every other. For n bundled facts of average arity k with random entity vectors, a union bound over cleanup failures gives a reliable capacity of

$$\text{Capacity} \sim \frac{d}{k \log(1/\delta)},$$

where δ is the acceptable per-query failure rate: each stored fact contributes crosstalk of variance $\Theta(1/d)$, and requiring the target to dominate n competitors with probability $1-\delta$ yields the stated bound. For $d = 8192$, $k = 2$, $\delta = 0.01$ this is ≈ 900 facts per bundle. Because interference grows with n, we use separate storage, which decouples encoding quality from knowledge-base size and admits ANN indexing for sublinear query time.

D Backward-Chaining Details

We implemented goal-directed proof search (backward chaining) via SLD resolution adapted to approximate VSA unification. To prove a goal g we first attempt to unify it with facts in the KB using VSA similarity (base case); if that fails we select rules whose head matches g, unify with the head, and recursively prove the body atoms (recursive case). Proven goals are cached (tabled) to prevent infinite loops in recursive rules, and search depth is bounded to guarantee termination. Unlike classical unification, which is exact, VSA unification is approximate: we build consistent substitutions whenever similarity exceeds threshold 0.5, handling variables on both goal and fact sides.

We tested backward chaining on the family tree with recursive `ancestor` rules; all 14 test cases pass. Ground facts resolve (proving `parent(alice,bob)` retrieves from the KB at similarity 0.95). Single-variable queries resolve (`parent(alice,X)` finds both children at similarity > 0.9). Recursive rules resolve (our running `ancestor(alice,carol)` via `ancestor(X,Z):-parent(X,Y),ancestor(Y,Z)` succeeds through the 2-hop chain `alice`→`bob`→`carol`). Tabling prevents divergence (the pathological `loop(X):-loop(X)` terminates). On grandparent queries, forward and backward chaining return the same answers with overlapping result sets, confirming semantic equivalence under approximate matching.

Table 6: Forward versus backward chaining.

Strategy	Best for	Cost
Forward	Bulk queries, large KB	Derives unused facts; memory
Backward	Sparse goals, deep recursion	Re-computation; needs cache

E Full Dimension Ablation

Table 7: Dimension ablation (hybrid + cancellation): mean retrieval similarity ± standard deviation by arity, seed 42, 100 queries per cell. Retrieval accuracy is 100% in every cell.

Dim	Binary	Ternary	Quaternary
512	0.926 ± 0.009	0.697 ± 0.030	0.502 ± 0.036
1024	0.926 ± 0.007	0.700 ± 0.018	0.498 ± 0.023
2048	0.924 ± 0.005	0.695 ± 0.015	0.499 ± 0.017
4096	0.924 ± 0.003	0.696 ± 0.011	0.499 ± 0.012
8192	0.924 ± 0.002	0.696 ± 0.007	0.500 ± 0.009
16384	0.924 ± 0.002	0.695 ± 0.005	0.500 ± 0.006

Across all three arities, retrieval accuracy is 100% at every dimension from 512 to 16384, and the *mean* similarity is essentially constant within each arity (set by the cancellation fraction $m/(k-1)$, not by d), reproducing the encoding-comparison values of the main paper (0.92, 0.70, 0.50). What dimension controls is *reliability*: the standard deviation of the similarity shrinks as $1/\sqrt{d}$ (e.g., binary $0.009 \to 0.002$ from $d = 512$ to 16384; a 4× reduction over a 32× span in d), the same scaling as the bind/unbind error $\epsilon \sim 1/\sqrt{d}$. Encoding cost meanwhile rises from ≈ 0.4 to 1.5 ms/fact and per-fact memory doubles with each doubling of d. We therefore adopt $d = 8192$ for a comfortable reliability margin and capacity headroom on larger knowledge bases; smaller dimensions already suffice at these benchmark sizes.

F Real Knowledge-Graph Evaluation

To test the encoding and reasoning beyond synthetic benchmarks, we evaluate on three standard neuro-symbolic knowledge graphs (MINERVA splits (Das et al., 2018)): **Kinship** (104 entities, 25 relations), **UMLS** (135 entities, 46 relations), and **Countries** S1–S3 (Bouchard et al., 2015) (271 entities, 2 relations; designed to test multi-hop rule reasoning). Entities and relations are assigned random hypervectors by id (no learned embeddings). We use two protocols.

Protocol A: argument recovery at real vocabulary scale. For each test triple (h, r, t) we encode the fact and use SIC to recover the queried argument (tail with h known, head with t known), then clean up against the *full* entity codebook, reporting filtered Hits@k and MRR. This extends the dimension ablation from ∼20 synthetic entities to 104–271 real entities. Recovery is perfect on all three (Table 8): the encoding and SIC decoding rank the correct entity first even when the codebook is 5–13× larger, at ≈ 14 ms/query with an exhaustive cleanup scan.

Table 8: Protocol A: argument recovery at real KG scale (hybrid + cancellation), $d = 8192$.

Dataset	$\lvert E\rvert$	$\lvert R\rvert$	Hits@1	Hits@10	MRR
Kinship	104	25	1.00	1.00	1.00
UMLS	135	46	1.00	1.00	1.00
Countries S1	271	2	1.00	1.00	1.00

Protocol B: multi-hop rule reasoning (Countries). Countries holds out `locatedIn`(country, region) test edges recoverable via two rules: transitivity, $\texttt{locatedIn}(X, Z) \leftarrow \texttt{locatedIn}(X, Y), \texttt{locatedIn}(Y, Z)$,

and a neighbor rule, $\texttt{locatedIn}(X, Z) \leftarrow \texttt{neighborOf}(X, Y), \texttt{locatedIn}(Y, Z)$. We insert the graph facts, forward-chain, and rank the true region (filtered) for each test query. On **S1**, which requires only transitivity, VSA-SIC + chaining is **perfect (MRR = Hits@1 = 1.00)**, deriving the needed transitive edges and recovering all 24 test triples.

An honest limitation: multi-rule composition and the vector-join. S2 and S3 require composing the neighbor rule with the transitive closure. Their answers remain derivable (a symbolic reachability check recovers 23/24), but our forward-chaining engine does not reach them within a 5-minute budget. The cause is instructive and specific to the approximate substrate: this reasoner performs each join by an $O(n)$ *vector-similarity scan* over stored facts (unbind, compare, threshold) rather than an $O(1)$ indexed lookup, and novelty-checks each derived candidate against the whole growing knowledge base. A single-rule closure (S1) is cheap, but the two-rule composition generates many candidates over several fixpoint iterations, and the similarity-scan cost compounds. A classical indexed Datalog solves S2/S3 in milliseconds. This exposes a real tension in the "approximate joins by retrieval" premise: vector-retrieval joins avoid relational *enumeration* blowup on large single-predicate retrieval, but incur their own cost on deep multi-rule composition. The bottleneck is the join and novelty check, not the SIC decoding, which is exact enough to solve S1 and to recover every argument in Protocol A. Approximate-nearest-neighbor indexing of the novelty check (which we use conceptually for retrieval but not in chaining) would improve the constants; the $O(n)$ similarity join is inherent to the design. We therefore scope the reasoning claim to shallow, single-rule composition and large-scale retrieval, and leave scalable multi-rule chaining to future work.

Baselines and positioning. On Countries the standard metric is AUC-PR. For context, the numbers reported by Rocktäschel & Riedel (2017) are, for S1/S2/S3: ComplEx (Trouillon et al., 2016) 99.4/88.0/48.4 and the neuro-symbolic prover NTPλ 100/93.0/77.3. The pattern is diagnostic: every method solves S1, but on S3 the non-compositional embedding model (ComplEx) collapses toward chance (48%) while the rule-learning NTP holds (77%). Our method also solves S1 exactly; its S2/S3 limitation is one of chaining *performance*, not expressivity (the answers are derivable). We caution that these baseline numbers are AUC-PR from prior work and not a same-metric head-to-head with our filtered Hits@k. On Kinship and UMLS, learned rule miners such as Neural LP (Yang et al., 2017) target held-out *link prediction* (generalization); our Protocol A instead measures encoding fidelity (recovery of stored facts), a complementary property, so the two are not directly comparable.

G Unlock Experiments: Protocols and Extended Data

This section gives the full protocols and extended data for the three unlock experiments of Section 7.7. All use $d = 8192$, seed 42, on Kinship and UMLS.

Open-vocabulary, training-free reasoning. We hold out a random 30% of entities (seed 42); a transductive ComplEx model (Trouillon et al., 2016) is trained only on facts whose head and tail are both retained, and we evaluate on the facts that involve a held-out entity. VSA needs no training phase (a held-out entity is simply a fresh random vector), so it encodes and decodes these facts directly, whereas ComplEx has no embedding for an unseen entity. VSA-SIC recovers their arguments at Hits@1 = 1.00 ($\sim$10–16 ms/fact, no training); ComplEx cannot assign a score to an unseen entity at all (Hits@1 = 0), and incorporating the new entities requires a full retrain (66 s on UMLS, 137 s on Kinship). The main-text Table 5 reports these numbers.

Robustness with calibrated confidence. We corrupt each stored fact hypervector with additive noise of relative magnitude η ($\mathbf{f}' = \text{normalize}(\mathbf{f} + \eta\|\mathbf{f}\|\mathbf{u})$, $\mathbf{u}$ a random unit vector) and recover the tail. Recovery stays at Hits@1 = 1.00 up to $\eta = 2$ (noise twice the signal), then degrades *gracefully* to chance by $\eta = 16$; the retrieval similarity tracks the corruption as a confidence signal (0.92 at η=0, 0.16 at η=1, 0.03 at η=8). An exact or hash-indexed store fails on *any* corruption ($\eta > 0$).

Multi-constraint conjunctive queries. A conjunctive query seeks entities satisfying several relational constraints at once, e.g. find X with $r_1(X, a) \wedge r_2(X, b)$. We answer it by retrieving the candidate set for each

Table 9: Graceful degradation under noise (Kinship). VSA Hits@1 vs. noise magnitude η; similarity is the calibrated confidence.

η	0	0.5	1	2	4	8	16
VSA Hits@1	1.00	1.00	1.00	0.99	0.92	0.31	0.06
Confidence	0.92	0.33	0.16	0.08	0.04	0.03	0.03
Exact H@1	1.00	0	0	0	0	0	0

constraint via SIC and intersecting by combined similarity, and score against the exact relational answer. Over 100 sampled conjunctive queries per dataset, VSA-SIC attains precision = recall = $F_1 = 1.00$ on both Kinship and UMLS: the multi-constraint joins that unreliable decoding previously collapsed to baseline are now exact.

H Experimental Details

Environment. Experiments ran on a commodity CPU-only configuration (Intel Core i7-10700K, 8 cores, 32 GB RAM), deliberately without GPU acceleration to demonstrate feasibility on standard hardware. The software stack was Python 3.11 with NumPy and JAX for vectorized array operations, the VSAX library for VSA primitives, and Matplotlib for figures. Results were verified on both Linux and Windows.

Reproducibility. The main runs fix the random seed at 42; entity and predicate vectors are sampled once from the unit sphere in $\mathbb{C}^{8192}$ and reused across runs, so results are deterministic up to floating-point differences (± 0.03). For the encoding-comparison error bars (Table 1) we additionally run seeds 41–50. Across those seeds the mean similarity is stable: the largest per-cell standard deviation is $\sigma \approx 0.03$ (role-filler, binary), and the permutation-based encodings (shift, hybrid, hybrid+cancel) are essentially seed-invariant in the mean ($\sigma \approx 0.00$), because their decoding uses only exact operations.

Hyperparameters. We use FHRR at $d = 8192$ (bind/unbind error $\epsilon \approx 0.05$). Retrieval thresholds are arity-dependent: 0.85 (binary), 0.60 (ternary), 0.50 (arity ≥ 4), set on held-out validation queries. Cleanup retrieves the top-10 nearest entity vectors by similarity. Forward chaining uses a beam width of 50, a maximum depth of 5 hops, and a novelty threshold of 0.90 to suppress near-duplicate derived facts. For the knowledge-graph reasoning experiment the join enumeration cap and beam are raised to exceed the per-predicate fact count.