SAME KNOWLEDGE, DIFFERENT GRAPH: KNOWLEDGE-GRAPH FOUNDATION MODELS ARE NOT SCHEMA-INVARIANT

Anonymous authors
Paper under double-blind review

ABSTRACT

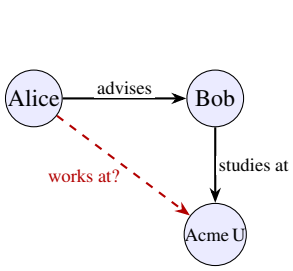
Knowledge-graph foundation models (KGFMs) are designed to be equivariant to the renaming of entities and relations, a property that supports transfer to unseen vocabularies. We ask whether they are also invariant to how the same knowledge is written down, for instance, as a direct edge, a reified event node, or a Freebase-style mediator. These encodings are not isomorphic, yet each can be made answer-preserving for a declared family of queries. We formalize such *schema rewrites* and evaluate KGFMs with a context-only protocol in which the background relations outside a fold are rewritten while the held-out queries and their candidates stay fixed. Under reification of the evidence, ULTRA's mean reciprocal rank falls by 17–55% relative to the original view on six of seven benchmark graphs, and its top-1 answer changes on 60–90% of queries. On CoDExSmall, DBpedia100k, and WN18RR, where deleting the rewritten facts measurably hurts, re-encoding them costs 0.8–1.3 times as much as deleting them. A Freebase-style expansion of the collapsed mediator relations of FB15k-237 costs 17%. Two further message-passing models behave similarly. We relate the failure to a receptive-field idealization of the entity encoder and to a removable query-conditioning asymmetry in ULTRA. Among remedies, a learned consistency objective does not consistently improve over augmentation. A structural normalizer with a relation-purity test, when the motifs in use are recognized, recovers near-original MRR and high top-1 agreement on every rewrite we constructed, with residual discrepancies where motifs have low support. It leaves the eight untouched graphs we tested unchanged; incomplete motif recognition remains its main open failure.

1 INTRODUCTION

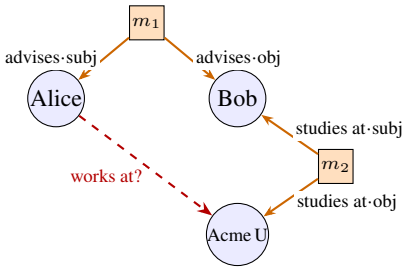
A knowledge graph (KG) stores facts as edges between entities, and the same fact can be stored in several ways: "Alice works at Acme U" can be one edge worksAt(Alice, Acme U), the inverse edge employs(Acme U, Alice), or, by *reification*, an event node f with role edges subject(f, Alice) and object(f, Acme U); Freebase (Bollacker et al., 2008) used a fourth form, a compound-value node that connects Alice to Acme U and also carries her job title and start date. We call the choice among such forms the *schema* of the graph. Real graphs mix schemas: 144 of the 237 relations of FB15k-237 (Toutanova & Chen, 2015) are two-step Freebase mediator paths collapsed into single edges, and Wikidata (Vrandečić & Krötzsch, 2014) stores each claim as a triple and as a statement node.

Knowledge-graph foundation models (KGFMs) such as ULTRA (Galkin et al., 2024a), TRIX (Zhang et al., 2025), MOTIF (Huang et al., 2025) and FLOCK (Kim et al., 2026) are trained once and applied to graphs with new entities and relations. They are designed to be *equivariant*, so that renaming nodes or relations does not change their answers, a property that supports transfer to unseen vocabularies without guaranteeing accuracy there. Two encodings of the same knowledge are not related by any renaming (reification adds nodes and relations and lengthens paths), so equivariance says nothing about whether a KGFM answers the same under both. We call this stronger property *schema invariance* and ask: does a KGFM give the same answer when only the encoding of its evidence changes?

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Observed context G and held-out query. Evidence path: 2 facts, 2 hops. Candidates: {Alice, Bob, Acme U}.



(b) Rewritten context $T_2(G)$: each fact becomes a mediator m_i (never a candidate) with two role edges. Per fact: $1 \to 2$ hops; evidence path: $2 \to 4$ hops. Query and candidates unchanged.

Figure 1: The context-only protocol on a toy context (solid: observed context; dashed: held-out query; amber: mediators created by the rewrite). The illustrative rule $\textit{advises}(x, y) \wedge \textit{studies at}(y, z) \Rightarrow \textit{works at}(x, z)$ makes the two context facts evidence for the query; benchmark graphs carry no such rules. One frozen model scores both views over the same candidates; per query, we compare filtered MRR and top-1 agreement, with two reference conditions on the same queries (rewritten relations deleted; the same number of introduced nodes and edges wired at random). Fig. 4 shows the same context under four encodings.

To answer it, we define *answer-preserving schema rewrites* relative to a declared query family and evaluate with a *context-only* protocol (Fig. 1): relations are partitioned into folds; in each fold the background relations outside the fold are rewritten while the held-out queries, their candidates and their filtered set are unchanged; and every admitted view is checked by symbolic execution. Under reification of the background evidence, ULTRA's MRR falls by 17–55% relative to the original view on six of seven graphs, and its top-1 answer changes on 60–90% of queries; on CoDExSmall, DBpedia100k, and WN18RR, where deleting the rewritten facts measurably hurts, re-encoding them costs 0.8–1.3 times as much as deleting them. A Freebase-style expansion of the collapsed mediator relations of FB15k-237 costs 17%, three to four times the cost of deletion, and for conjunctive and disjunctive queries, re-encoding costs four times what deletion costs. We relate these observations to a receptive-field idealization of the entity encoder (Thm. 1) and to a removable query-conditioning asymmetry in ULTRA (Prop. 1); MOTIF and TRIX behave similarly, and FLOCK's random walks lose far less consistency at the budget we tested. Among remedies, a learned consistency objective does not consistently improve over augmentation on unseen rewrite families. A blind structural normalizer with a relation-purity test, when the motifs in use are recognized, recovers near-original MRR and high top-1 agreement on every rewrite we constructed, with residual discrepancies where motifs have low support. It left all eight untouched graphs we tested unchanged; incomplete motif recognition is its main open failure.

In summary, our contributions are (i) a formal definition of schema invariance for KGFMs and SCHEMASHIFT, a benchmark whose rewrites are validated by execution and whose protocol keeps the query, its candidates and its filtered set fixed (§3); (ii) an empirical diagnosis on seven KGs, three message-passing backbones and one walk-based backbone, with a Freebase-style encoding and complex logical queries, together with a depth idealization and a verified conditioning mechanism (§4, §5); and (iii) an evaluation of learned and structural remedies that reports where each works and fails, with a conditional guarantee for the normalizer (§6).

2 Related Work

Inductive link prediction first generalized to unseen entities (Teru et al., 2020; Zhu et al., 2021), then to unseen relations by reasoning over a graph of relations (Lee et al., 2023; Gao et al., 2023; Galkin et al., 2024a). ULTRA (Galkin et al., 2024a) is the backbone we study most closely. Related KGFMs differ in what they add: TRIX (Zhang et al., 2025) and MOTIF (Huang et al., 2025) increase expressivity, the latter by encoding relational motifs; KG-ICL (Cui et al., 2024) reasons in context from prompt graphs; HYPER (Huang et al., 2026) handles higher-arity relations; UltraQuery (Galkin et al., 2024b) answers logical queries; and FLOCK (Kim et al., 2026) replaces message passing by

anonymized random walks. These models are built on node–relation equivariance, whose guarantees concern renamings of a fixed graph rather than re-encodings of it (Finkelshtein et al. (2025) study symmetries of graph foundation models more generally). Huang et al. (2025) and Zhang et al. (2025) characterize which structurally *different* graphs a KGFM can distinguish; our question is whether its predictions are consistent across structurally different but answer-preserving encodings of the *same* knowledge. FLOCK's observation is that equivariance can be too strong (isomorphic relations can have opposite meanings); ours is the converse.

Reification, compound-value types, and qualifier statements are standard ways of storing n-ary facts in a binary graph (Bollacker et al., 2008; Wen et al., 2016; Vrandečić & Krötzsch, 2014), consumed directly by hyper-relational models (Galkin et al., 2020; Huang et al., 2026); to our knowledge, whether a KGFM's answer depends on the encoding chosen has not been measured. Learning an invariance from transformed inputs is well understood when the transformations form a group acting on a fixed space (Chen et al., 2020); schema rewrites map between spaces of different cardinality (Prop. A.2), so that the theory does not apply directly, and we test augmentation and consistency objectives empirically. Adversarial modifications of link predictors (Pezeshkpour et al., 2019) add or remove observed facts; we study rewrites designed to preserve the answers of a declared query family.

3 Schema Rewrites and Schema Invariance

3.1 Knowledge graphs, queries and link invariance

A knowledge graph is a triple $G = (V, R, E)$ with entities V, relation types R and directed labeled edges $E \subseteq V \times R \times V$. In link prediction the model observes a *context graph* G_{obs} (the training edges) and answers *queries* $q = (h, r, ?)$ or $(?, r, t)$ built from held-out test triples absent from G_{obs}, returning a score vector $s_f(G_{\text{obs}}, q) \in \mathbb{R}^C$ over a fixed *candidate set* $C \subseteq V$; the held-out entity is the *reference answer*, and the filtered protocol removes from the ranking every other candidate known to be correct from the *filtering graph* F (all splits). Let $\mathcal{K}_{n,m}$ be the set of KGs with n entities and m relations (Kim et al., 2026), and an *isomorphism* a pair of bijections (π, ϕ) on V and R mapping one edge set onto another. A KGFM f is *link-invariant* if its scores are unchanged under every isomorphism within $\mathcal{K}_{n,m}$, exactly or in distribution; ULTRA, TRIX, MOTIF and FLOCK are built to satisfy this.

3.2 Schema rewrites and schema invariance

A rewrite changes the encoding of a graph and comes with rules for translating queries and answers.

Definition 1 (Schema rewrite)**.** *A schema rewrite is a map $T : \mathcal{K}_{n,m} \to \mathcal{K}_{n',m'}$ together with an injective entity map $\alpha_T : V \to V'$ and a query translator τ_T that maps each query of a declared family $\mathcal{Q}$ to a path query on $T(G)$. Let $\mathrm{Ans}_G(q) \subseteq V$ be the set of entities reached from the anchor by executing q on the edges of G (for $q = (h, r, ?)$, the r-successors of h; a path query is executed hop by hop). The rewrite is* answer-preserving *for $\mathcal{Q}$ if $\mathrm{Ans}_{T(G)}(\tau_T q) = \alpha_T(\mathrm{Ans}_G(q))$ for every $q \in \mathcal{Q}$, the right-hand side being the image under α_T.*

These answer sets are defined over observed edges only and say nothing about a benchmark query's held-out reference answer or the illustrative rule of Fig. 1; in the link-prediction experiments, the models answer one-hop queries, so there we use only rewrites whose τ_T returns one (every context-only rewrite and global inversion); the complex-query extension (§5.3) applies context-only rewrites to UltraQuery's logical queries.

Definition 2 (Schema invariance)**.** *A deterministic model f is* score-invariant *to T on $\mathcal{Q}$ if $s_f(T(G), \tau_T q)[\alpha_T(c)] = s_f(G, q)[c]$ for every $q \in \mathcal{Q}$ and $c \in C$. A stochastic model is score-invariant* in distribution *if, for every q, the joint distribution of the aligned score vector $\big(s_f(T(G), \tau_T q)[\alpha_T(c)]\big)_{c \in C}$ equals that of $\big(s_f(G, q)[c]\big)_{c \in C}$. For a deterministic model with consistent tie-breaking, score invariance implies* ranking invariance *and hence* top-1 agreement*, so agreement 1 is necessary but not sufficient; for a stochastic model two independently sampled predictions can disagree even under invariance in distribution, which is why §5.4 compares agreement against the model's own resampling reference.*

The rewrite primitives are (Fig. 4): *renaming* T_0 (a control); *inversion* T_1, which replaces every (h, r, t) by (t, r, h), is applied *globally*, and whose τ_{T_1} flips each query's direction; *reification* T_2, which replaces every (h, r, t) of a chosen relation subset by a fresh node f and role edges $(f, \arg_0^r, h)$, $(f, \arg_1^r, t)$, with roles specific to r or a shared subject role; and *path expansion* T_3, which writes the fact as (h, r_{src}, f), (f, r_{tgt}, t), the shape of Freebase's compound-value paths. A *mixed* schema applies different primitives to disjoint relation slices; T_2^d applies reification d times. The *Freebase-style expansion* takes the 144 relations of FB15k-237 whose names are collapsed property pairs $p_1.p_2$ and writes each such fact as (h, p_1, m), (m, p_2, t), with one fresh mediator per fact and one relation per distinct Freebase property, shared across relations as in Freebase; the original mediator identities and qualifiers, absent from FB15k-237, are not recovered.

3.3 The context-only protocol and its metrics

Except for the two global controls, every rewrite is *context-only*. Relations are partitioned into $K = 5$ folds, deterministically balanced by test frequency. In fold k every relation outside the fold is rewritten, and we evaluate held-out triples whose relation lies inside the fold, in both directions; the folds partition the full test set, and on large graphs a seeded per-fold cap on test triples is applied (Appendix C). Across views the query is therefore the same one-hop link prediction, τ_T is the identity, C is the set of original entities (introduced nodes are never candidates), and the filtered set is $\alpha_T(F)$; each rewritten fact is replaced by a fixed pattern (two role edges through one introduced node) with a recorded relation map, and α_T is the identity on entities.

Before a view is admitted, a symbolic executor runs every evaluation query on G_{obs} and its translation on $T(G_{\text{obs}})$ and requires equal answer sets with no introduced node among them. Since query relations are never rewritten, this verifies that the query-relation edges and the entity identities survived the rewrite; the encoding of the rewritten background facts rests on its construction and an edge-level round-trip check on a small graph (Appendix B). Held-out facts are in neither graph.

Three quantities are reported. *Accuracy* is filtered MRR over C, with relative loss $\Delta = (\text{MRR}_{\text{orig}} - \text{MRR}_{\text{view}})/\text{MRR}_{\text{orig}}$ (positive for a loss) and δ for an absolute signed change. *Prediction consistency* is *equivalent-view consistency* EVC, the fraction of queries whose top-1 candidate is the same in the original and the rewritten view. Scoring the same graph twice with a deterministic model gives EVC 0.999; for a stochastic model the reference EVC_{ref} is its own agreement between two re-scorings of the original graph, and the *normalized consistency deficit* $\text{NCD} = (\text{EVC}_{\text{ref}} - \text{EVC})/\text{EVC}_{\text{ref}}$ is the agreement loss relative to that reference (it can be negative, and a value near zero means agreement at the resampling level, not equal output distributions). *The deletion comparison* uses two reference conditions on the same queries: *deletion* removes the rewritten relations, and the *schema cost ratio* $\text{SCR} = (\text{MRR}_{\text{orig}} - \text{MRR}_{\text{view}})/(\text{MRR}_{\text{orig}} - \text{MRR}_{\text{deleted}})$ compares re-encoding with deletion where deletion measurably hurts; *matched noise* deletes the relations and adds the same number of introduced nodes and edges with the same mediator degree profile, wired to random entities, controlling for node and edge counts and mediator degree but not every topological difference.

4 What Link Invariance Does and Does Not Cover

ULTRA answers $(h, r, ?)$ in two stages (Appendix C.1): a *relation encoder* runs conditional message passing, seeded at the query relation, over a graph whose nodes are relation types and whose edges record the four ways two relations can share an entity; an *entity encoder* then runs $L = 6$ layers of conditional message passing over the entity graph, seeded at the anchor h with those relation representations as edge features, and scores every entity from its final state. Renaming is an isomorphism within $\mathcal{K}_{n,m}$, so every link-invariant model is invariant to T_0 (Prop. A.1); the remaining rewrites are not.

Proposition 1 (Global inversion and ULTRA's query conditioning)**.** *Under T_1 the inverse-augmented edge multiset of $T_1(G)$ equals that of G under the relabeling $\sigma(r) = r^{-1}$, and ULTRA's relation-interaction graph of $T_1(G)$ is σ applied to that of G with the four interaction types preserved. ULTRA as released is nevertheless not T_1-invariant: the relation encoder is seeded at the batch relation r while head prediction reads the query vector at r^{-1}, so the same question posed in the two directions is computed under different conditioning. Seeding the encoder at the relation actually read makes the two computations identical up to σ.*

Reification of k facts maps $\mathcal{K}_{n,m} \to \mathcal{K}_{n+k,m'}$, so no isomorphism $G \simeq T_2(G)$ exists and link invariance does not constrain $f(T_2(G))$ (Prop. A.2); for a model that augments E with inverse relations, T_2 and T_3 yield identical outputs (Prop. A.3).

Theorem 1 (Receptive field under composed reification)**.** *Let f score candidates with an L-layer conditional message-passing entity encoder in which the anchor h carries the boundary condition, all other initial states are zero, and messages, aggregation and updates are* zero-insensitive*: a message from a zero state is zero, an aggregate is unchanged when zero messages are added or removed (the aggregate of nothing being zero), and the update of a zero state with a zero aggregate is zero. Sum aggregation qualifies and a mean does not (added zero neighbors change its denominator); NBFNet with sum aggregation and no bias or normalization terms is an example. Relation representations and all other encoder inputs are held fixed across the graphs compared. For a rewrite T that replaces each edge e by a path of $2^{d(e)}$ mediator-linked edges, let $\ell_T(h, v) = \min_P \sum_{e \in P} 2^{d(e)}$ over paths P from h to v in the inverse-symmetrized G. If $E' \subseteq E$ is a set of edges all of whose endpoints satisfy $\ell_T(h, \cdot) > L$ and $G^- = G \setminus E'$, then for every parameter setting $s_f(T(G), \tau_T q) = s_f(T(G^-), \tau_T q)$ on all candidates; hence f cannot be both score-invariant to T on $\{G, G^-\}$ and sensitive to E' on G. If every edge on a path of length ℓ is rewritten d times and no lighter path exists, $\ell_T = 2^d \ell$.*

Proof sketch. We compare the two executions, on $T(G)$ and $T(G^-)$, layer by layer: by induction, after layer t every node state is identical in both and zero outside the t-hop ball of h, because the only nonzero messages come from nodes inside the $(t-1)$-hop ball along edges that cannot belong to E', and zero-insensitive aggregation ignores the rest. Mediators have degree two, so distances between original entities in the symmetrized $T(G)$ equal ℓ_T (Appendix A). □

The theorem is an explanatory idealization: the released encoders of ULTRA, MOTIF and TRIX include bias and normalization terms and, in ULTRA, a relation encoder computed from the whole graph, so their states outside the anchor's ball are neither zero nor graph-independent, and we do not claim that the theorem constrains a separable component of those models. It concerns dependence, not accuracy, and implies neither a saturation depth nor that shortening paths must help; the depth observations of §5 are consistent with it, not tests of it (Appendix A).

5 Are KGFMs Schema-Invariant?

We use the released checkpoints of four KGFMs (Appendix C.1) without retraining, mainly ULTRA's `ultra_3g`, which reproduces the published zero-shot numbers (Appendix D); the KGs are CoDExSmall (Safavi & Koutra, 2020), WN18RR (Dettmers et al., 2018), DBpedia100k (Ding et al., 2018), AristoV4, Hetionet (Himmelstein et al., 2017), ConceptNet100k (Malaviya et al., 2020) and FB15k-237; WN18RR and FB15k-237 are in every backbone's pretraining mixture and are transductive rather than zero-shot.

5.1 Reification of the evidence, with controls and reference conditions

Renaming (T_0) leaves every number unchanged (EVC 0.999–1.000; Prop. A.1) and path expansion (T_3) is identical to reification to the last reported digit (Prop. A.3); both serve as controls.

Table 1 and Fig. 5 give the diagnosis on six graphs. On CoDExSmall, WN18RR and DBpedia100k, where deleting the rewritten relations measurably hurts, re-encoding costs 0.82–1.32 of what deletion costs: SCR is 0.98 on CoDExSmall, 0.82 on WN18RR, where some of the re-encoded evidence is still used, and 1.26 and 1.32 on the two DBpedia100k seeds (each computed from that seed's pooled MRRs; Table 1 lists their mean), where the model scores lower with the facts rewritten than with them removed; on FB15k-237 (Table 2) the per-seed ratios are 3.1–4.2. The matched-noise condition argues against node and edge count as the explanation: on DBpedia100k, the reified view (0.182) is level with the random structure of the same size and shape (0.177). Where deletion *helps* (AristoV4, Hetionet, ConceptNet100k), SCR is withheld because its precondition fails, and MRR measures accuracy but not the value of the evidence; the consistency metric is the most stable quantity in the table, EVC 0.10–0.39 on all six graphs against a reference of 0.999, i.e., the top-1 answer changes on 60–90% of queries under a rewrite that preserves every answer set.

Table 1: Context-only reification across six KGs with ULTRA (`ultra_3g`, five relation folds pooled). n: directional queries per seed; $\times 2$: mean of two seeds; ‡: n is the whole test split, so a second seed changes only fresh-node identifiers and reproduces the run. Δ: relative MRR loss; "Deleted": MRR with the rewritten relations removed; SCR: mean of the per-seed schema cost ratios, withheld (†) where deletion does not measurably hurt. FB15k-237 is in Table 2.

Dataset	n	Orig. MRR	Reified	Δ	EVC	Deleted	SCR
AristoV4†	6000×2	0.183	0.151	+17.7%	0.099	0.193	–
CoDExSmall	3656‡	0.472	0.355	+24.6%	0.389	0.353	0.98
ConceptNet100k†	2400‡	0.061	0.154	-150.8%	0.291	0.092	–
DBpedia100k	10000×2	0.396	0.180	+54.5%	0.252	0.229	1.29
Hetionet†	5000×2	0.244	0.154	+36.6%	0.208	0.264	–
WN18RR	4018×2	0.375	0.240	+36.0%	0.220	0.210	0.82

Table 2: FB15k-237: the 144 collapsed Freebase relations expanded into a Freebase-style mediator encoding (ULTRA `ultra_3g`, five folds, $n = 6{,}000$ directional queries per seed; transductive). Δ is relative to the collapsed view of the same seed; the last row applies the two-motif normalizer with the purity test (all variants in Table 7).

View	seed 0			seed 1		
	MRR	Δ	EVC	MRR	Δ	EVC
original (collapsed, as shipped)	0.365	–	–	0.363	–	–
Freebase-style mediator expansion	0.304	+16.8%	0.408	0.299	+17.8%	0.402
synthetic reification, same relations	0.296	+19.0%	0.398	0.290	+20.1%	0.388
control: relations deleted	0.345	+5.4%	0.472	0.346	+4.8%	0.480
control: matched-structure noise	0.225	+38.4%	0.229	0.227	+37.6%	0.247
expansion + normalizer (two motifs, purity)	0.365	+0.0%	1.000	0.363	+0.0%	1.000

Table 3: Three message-passing KGFMs under context-only reification, from their shipped checkpoints (seed 0, five folds pooled; original and reified MRR and per-fold query caps in Table 6). Δ: relative MRR loss under reification; EVC: top-1 agreement; "inv. EVC": agreement under global inversion. WN18RR is in each model's pretraining mixture. †: deletion helps this backbone on this graph, so SCR is withheld. MOTIF did not fit in host memory on AristoV4.

Graph	ULTRA			MOTIF			TRIX		
	Δ	EVC	inv. EVC	Δ	EVC	inv. EVC	Δ	EVC	inv. EVC
CoDExSmall	+24.6%	0.389	0.437	+13.4%	0.284	1.000	+7.8%	0.450	0.364
WN18RR	+35.7%	0.218	0.267	+24.6%	0.298	0.999	+18.6%	0.473	0.601
DBpedia100k	+54.3%	0.251	0.467	+26.2%	0.371	0.998	+18.0%	0.456	0.488
AristoV4	+18.0%†	0.097	0.282	–	–	–	-18.2%†	0.370	0.388

5.2 A Freebase-style encoding, and two further message-passing models

Table 2 gives the Freebase-style experiment. The expansion costs 16.8% and 17.8% relative MRR on the two seeds and our synthetic reification of the identical relations 19.0% and 20.1%; the two encodings differ by 2.2–2.3 percentage points of relative loss and by at most 0.014 in EVC (0.39–0.41), and each costs 3.1–4.2 times what deleting the relations costs, so the synthetic rewrite measures an effect that a Freebase-style encoding of real data also produces.

Table 3 repeats the protocol with MOTIF and TRIX. On CoDExSmall, WN18RR and DBpedia100k all three change their top-1 answer on 53–78% of queries and re-encoding costs 0.44–1.32 of what deletion costs; on CoDExSmall MOTIF loses less MRR than ULTRA yet more consistency (EVC 0.28 vs 0.39). Under global inversion, the models separate as Prop. 1 predicts: MOTIF, whose relation encoder is not seeded at the query relation, keeps EVC 0.998–1.000, whereas TRIX (0.36–0.60)

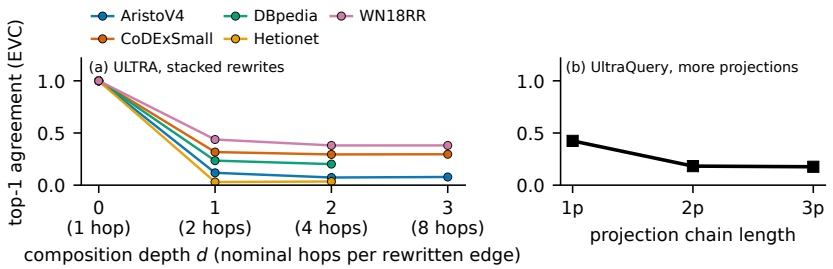


Figure 2: Top-1 agreement under (a) composed context reification with the frozen ULTRA checkpoint (fold 1, seed 0; axis: composition depth, nominal hops per rewritten edge in parentheses) and (b) UltraQuery on FB15k-237 logical queries with singly-reified evidence (pooled over two relation folds and two partition seeds; axis: number of projections, each running the full entity encoder; negation types are in Table 8). Both show a large drop at the first step and little further change, consistent with the idealization of Thm. 1 but not a test of it.

shares ULTRA's asymmetry (0.27–0.47). On AristoV4, where deletion helps every backbone, TRIX's MRR rises while its top-1 answer changes on 63% of queries.

5.3 Depth sensitivity and query conditioning

Most of the loss under composed reification occurs at the first level. Composing reification d times turns one rewritten edge into a nominal 2^d hops; on CoDExSmall (fold 1; fold-level MRR differs from the pooled value of Table 1) MRR falls from 0.799 at depth 0 to 0.527 at depth 1 and then changes little (0.496, 0.495), with EVC 0.32, 0.29, 0.30, and four further graphs show the same shape (Fig. 2a), consistent with the idealization of Thm. 1, which agreement alone does not test.

Under complex logical queries, the relative MRR loss is of the same size as for link prediction on the same graph, but its ratio to deletion is larger. Lifting the protocol to conjunctive and disjunctive queries with UltraQuery on FB15k-237 (Table 8; 22,184 evaluations) costs 16.8% relative MRR (link prediction under the Freebase-style expansion: 16.8–17.8%), with 25% of top-1 answers unchanged, whereas deleting the same evidence costs 4.3% and leaves 40%. The schema cost ratio is 4.4 as the mean of the four per-run ratios (2.6–6.6) and 3.9 from the pooled losses, against 3.1–4.2 per seed for link prediction. The two experiments use different models (UltraQuery versus ULTRA) and query sets, so the comparison does not isolate query complexity. Within the complex-query experiment, EVC is 0.42 for one projection and 0.18 for two and three (Fig. 2b), and 0.11 for projection with negation ("pin", Table 8).

Inversion sensitivity has an identified and removable cause, the conditioning asymmetry of Prop. 1: ULTRA's relation-interaction graph is preserved exactly under global inversion, yet the model is not invariant (Table 9; EVC 0.13–0.48, relative MRR loss up to 28%). Seeding the relation encoder at the relation actually read gives EVC 0.999–1.000 on all four graphs at inference, the level of the deterministic reference; with the shipped weights it changes original-view MRR by -36% to $+5\%$ (the checkpoint was never trained under this conditioning), and after matched fine-tuning under each conditioning the original-view difference over six fold/seed runs has mean -0.001, sd 0.017 (three of six bootstrap intervals exclude zero, in both directions), with EVC 0.996–1.000 under inversion (Table 10).

5.4 A walk-based model at a reduced budget

The walk-based model loses far less consistency under reification than ULTRA on the same queries. We compare the two with matched three-graph pretraining on five graphs (eight datasets/seed pairs; Table 11, Fig. 8); FLOCK is stochastic, so its agreement between two re-scorings of the original graph (0.40–0.79) is its NCD reference, listed with raw agreement and accuracy. Under reification FLOCK's NCD is 0.07–0.14 on four graphs and -0.02 on Hetionet, where ULTRA's is 0.62–0.91; on DBpedia100k FLOCK loses about half as much MRR under reification as under deletion (SCR 0.48–0.56) whereas ULTRA loses more (1.26), and under inversion its NCD is -0.02 to 0.03 everywhere.

Two qualifications apply: by the MRR-based cost ratio FLOCK is worse on CoDExSmall (3.2 vs 1.0), where deletion barely hurts it while reification costs it 15%; and FLOCK runs at a reduced walk budget, with absolute MRR on DBpedia100k, AristoV4 and Hetionet below ULTRA's and its published values (Appendix D).

6 Can the Model Learn It, or Should the Graph Be Normalized?

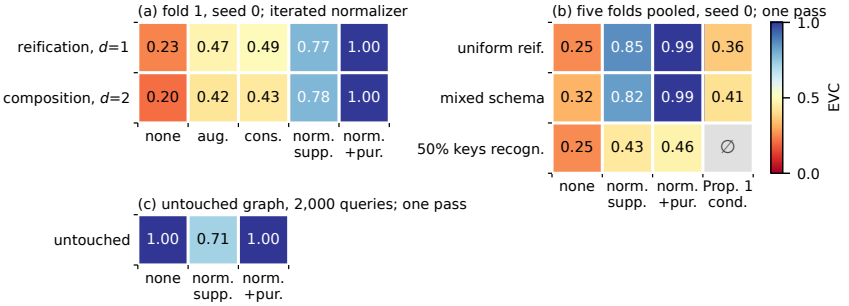


Figure 3: Top-1 agreement (EVC) on DBpedia100k for each remedy under each failure mode; within a panel every cell uses the same fold(s), query set and reference. (a) Fold 1, seed 0: learned columns are models fine-tuned on depth-1 reification (EVC compares each model's own two views); normalizer columns are the iterated normalizer (support-only stopped at the six-pass cap without converging; purity-filtered converged). (b) Five folds pooled, seed 0, frozen checkpoint, one pass. (c) Untouched graph, 2,000 queries, one pass. ∅: not applicable.

6.1 A learned objective does not consistently improve over augmentation

We fine-tune ULTRA on the original view plus a reified view under three objectives (Fig. 3a; Appendix C), *augmentation* (the model's own link-prediction loss on both views), *consistency* (adding a Jensen–Shannon term that asks the two views' answer distributions over identical candidates to agree) and *consistency with margin* (further asking a meaning-changing rewrite, the same mediators re-wired at random, to be farther from the original than the answer-preserving one), and evaluate on *held-out* rewrite families (unseen mixed and shared-role encodings). Over six-fold/seed runs on CoDExSmall, augmentation raises held-out MRR by $+0.021$ (mixed) and $+0.019$ (shared-role) on average, positive in five of six runs; consistency with margin adds a paired mean of $+0.001$ (mixed, sd 0.022) and $+0.004$ (shared-role, sd 0.018) (Table 13). On DBpedia100k (augmentation worth $+0.09$–0.12 MRR) the consistency term differs from augmentation by -0.004 to $+0.010$ over two seeds, and it sits below augmentation on unseen composition depths (Table 15). Fine-tuning on the rewritten view itself recovers most of the zero-shot gap, but a residual gap after matched fine-tuning remains on every run (0.026–0.039 MRR on CoDExSmall, 0.041–0.148 on DBpedia100k; bootstrap intervals exclude zero; Table 14).

6.2 Structural normalization, with a relation-purity test

The normalizer works in four steps. First, it identifies nodes that could be mediators from their edge pattern alone: a *reification motif* is a node with no incoming edge and two outgoing edges under a relation pair (a, b), a *path motif* is a node with one incoming edge under a and one outgoing edge under b. Detection uses graph structure alone, without a rewrite manifest or labels identifying introduced mediator nodes; separately, evaluation keeps the original candidate entities and maps query relations through the relation mapping the normalizer returns (Appendix B). Second, it requires repeated evidence that a pattern is a convention: a relation pair (a *key*) is accepted only if at least $s = 2$ nodes carry it. Third, it checks *relation purity* (below). Fourth, it collapses every accepted node, removing its edges and adding one edge with a fresh recovered relation between its two neighbors; composed rewrites are handled by repeating the pass until nothing collapses, with a six-pass cap (Alg. 1). On CoDExSmall, where no genuine node has a mediator's shape, the normalizer returns uniform, mixed and composed rewrites to within 0.001 MRR of the original, with EVC 0.989–1.00 for ULTRA and 0.94–1.00 for MOTIF and TRIX (Tables 17, 19).

DBpedia100k, by contrast, has genuine nodes with a mediator's shape, and there the support-only normalizer accumulates false positives: one pass collapses 1.6k genuine nodes (EVC 0.94) and the iterated version, stopped by the six-pass cap before converging, 5.5k (EVC 0.77–0.78 at both depths on both folds), while MRR moves only slightly in either direction (0.538 vs 0.531 on fold 1, 0.444–0.445 vs 0.448 on fold 2; Tables 15, 16); similar MRR conceals changed predictions. The purity test uses the fact that a relation created by a rewrite lives only on the mediators it created, whereas a genuine relation on mediator-shaped nodes also appears elsewhere: with π_r the fraction of all edges labeled r whose mediator-side endpoint is a candidate of an accepted key of the same motif family, a key (a, b) is kept only if $\min(\pi_a, \pi_b) \geq \tau$, with $\tau = 0.9$ throughout (Fig. 10).

Proposition 2 (Relation purity)**.** *Consider the normalizer above with thresholds s and $\tau \leq 1$, and a rewrite T whose created relations occur only on the mediators it creates, each carrying the degree signature of T's motif (all rewrites of §3 except T_0, T_1). (i) Every key that T creates has purity 1 and is accepted at any rewrite coverage if its relations were rewritten in at least s facts. (ii) A genuine key (a, b) is accepted only if at least a fraction τ of* all *edges labeled a, and of all labeled b, lie on candidates of the same motif family; on an untouched graph the normalizer is therefore the identity unless two genuine relations are each so confined to $\geq s$ mediator-shaped nodes with a common key. (iii) If no two accepted candidates are adjacent in any pass, each non-empty pass removes at least one node and adds none, so unbounded iteration terminates within $|V|$ passes; when only created keys are accepted this holds, and iteration on $T^d(G)$ returns the original graph up to relation renaming after d collapsing passes if every created key meets the support threshold.*

Part (ii) makes "false positive" a checkable condition; part (i) is why an unrecognized motif is untouched by the test; part (iii) does not cover passes that accept genuine keys (Appendix B). With the purity test, no node is collapsed on any of the eight untouched benchmark graphs tested (MRR within 0.0002 and EVC 0.999–1.000 on the five scored; Tables 21, 20).

On rewritten graphs the purity-filtered normalizer collapsed only created mediators (structural counts, Appendix B) and its iteration converged on every view; scored (Fig. 9), it returns DBpedia100k, WN18RR and Hetionet to within 0.003 MRR of the original with EVC 1.00 at every depth, matches the oracle-boundary normalizer on DBpedia100k within 0.0001 MRR (EVC 0.989–0.993; Table 18), and returns the Freebase-style expansion to the collapsed view's MRR to four decimals with EVC 1.000 on both seeds (Table 2), all agreements at the reported precision (Appendix B). The remaining disagreement coincides with mediators of single-fact relations that the support threshold keeps (74 on AristoV4, EVC 0.93; 27 on DBpedia100k), a proposed explanation that no ablation isolates.

Three failure modes remain: a normalizer whose library lacks the path motif recovers none of the Freebase-style expansion (Appendix B); *partial recognition*, a motif library covering only some conventions in use (simulated by withholding a random half of the accepted keys after the purity test), is the dominant open failure and costs 11–33% relative MRR on uniform and 6–21% on mixed views (Tables 17, 18; 4–16% across backbones); and adding the collapsed edge without removing the mediator is worse everywhere (Table 22).

7 Discussion and Conclusion

Link invariance guarantees consistency under renaming; it does not guarantee consistency under schema rewrites. Across the evaluated settings, three message-passing KGFMs change most of their top-1 answers under answer-preserving schema rewrites. For ULTRA, re-encoding costs 0.8–1.3 times as much as deleting the evidence on CoDExSmall, DBpedia100k, and WN18RR, and three to four times as much on FB15k-237. Within the evaluated conditions, we recommend normalizing known mediator conventions before reasoning: when the relevant motifs are recognized, purity-filtered normalization recovers near-original MRR and high top-1 agreement, with residual discrepancies from low-support motifs and incomplete recognition.

The study has limitations. It covers seven datasets, evaluates learned remedies on one backbone, and compares one walk-based model at a reduced budget. Theorem 1 is an idealization not shown to constrain the released models; Proposition 2 is conditional and does not cover the six-pass cap. Partial recognition, single-edge mediators, and higher-arity encodings remain open.

Reproducibility Statement

Appendix A gives complete proofs with all assumptions stated, and Appendix B gives the normalizer as pseudocode with its stopping conditions. Appendix C lists the datasets and splits, the fold construction, the seeds and what they control, the hyperparameters and training steps of every learned remedy, the walk budgets, the hardware and software versions, and the bootstrap procedure behind every reported interval. Appendix D records the pretrained checkpoints used, the reproduction of published numbers with their remaining discrepancies, and the evaluation safeguards applied to every run. Appendix E collects the remaining tables and figures, and Appendix F summarizes the scope of the study and what each control accounts for. The benchmark construction code, the normalizer, the evaluation procedure, and the saved predictions from which every reported number is computed will be released as supplementary material.

Ethics Statement

This work uses only publicly available benchmark knowledge graphs and released model checkpoints and involves no human subjects or personal data. Its findings concern the reliability of knowledge-graph reasoning systems: a model whose answers depend on the storage convention of its evidence could give different answers to the same question in deployments that mix conventions, for example, in a biomedical graph such as Hetionet. Making that dependence measurable is intended to reduce such risks. We see no direct path to misuse.

AI Use Statement

Generative AI tools assisted with language editing and polishing the manuscript. The authors reviewed the final text and take full responsibility for all scientific claims, analyses, and conclusions.

References

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: A collaboratively created graph database for structuring human knowledge. In *SIGMOD*, pp. 1247–1250, 2008. doi: 10.1145/1376616.1376746.

Shuxiao Chen, Edgar Dobriban, and Jane H. Lee. A group-theoretic framework for data augmentation. *Journal of Machine Learning Research*, 21(245):1–71, 2020. arXiv:1907.10905.

Yuanning Cui, Zequn Sun, and Wei Hu. A prompt-based knowledge graph foundation model for universal in-context reasoning. In *NeurIPS*, 2024. arXiv:2410.12288.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2D knowledge graph embeddings. In *AAAI*, 2018. arXiv:1707.01476.

Boyang Ding, Quan Wang, Bin Wang, and Li Guo. Improving knowledge graph embedding using simple constraints. In *ACL*, 2018. arXiv:1805.02408.

Ben Finkelshtein, İsmail İlkan Ceylan, Michael Bronstein, and Ron Levie. Equivariance everywhere all at once: A recipe for graph foundation models. In *NeurIPS*, 2025. arXiv:2506.14291.

Mikhail Galkin, Priyansh Trivedi, Gaurav Maheshwari, Ricardo Usbeck, and Jens Lehmann. Message passing for hyper-relational knowledge graphs. In *EMNLP*, 2020. arXiv:2009.10847.

Mikhail Galkin, Xinyu Yuan, Hesham Mostafa, Jian Tang, and Zhaocheng Zhu. Towards foundation models for knowledge graph reasoning. In *ICLR*, 2024a. arXiv:2310.04562.

Mikhail Galkin, Jincheng Zhou, Bruno Ribeiro, Jian Tang, and Zhaocheng Zhu. A foundation model for zero-shot logical query reasoning. In *Advances in Neural Information Processing Systems*, volume 37, 2024b. arXiv:2404.07198.

Jianfei Gao, Yangze Zhou, Jincheng Zhou, and Bruno Ribeiro. Double equivariance for inductive link prediction for both new nodes and new relation types. In *NeurIPS 2023 Workshop: New Frontiers in Graph Learning (GLFrontiers)*, 2023.

Daniel S. Himmelstein, Antoine Lizee, Christine Hessler, Leo Brueggeman, Sabrina L. Chen, Dexter Hadley, Ari Green, Pouya Khankhanian, and Sergio E. Baranzini. Systematic integration of biomedical knowledge prioritizes drugs for repurposing. *eLife*, 6:e26726, 2017. doi: 10.7554/eLife.26726.

Xingyue Huang, Pablo Barceló, Michael M. Bronstein, İsmail İlkan Ceylan, Mikhail Galkin, Juan L. Reutter, and Miguel Romero Orth. How expressive are Knowledge Graph Foundation Models? In *ICML*, 2025. arXiv:2502.13339.

Xingyue Huang, Mikhail Galkin, Michael M. Bronstein, and İsmail İlkan Ceylan. HYPER: A foundation model for inductive link prediction with knowledge hypergraphs. In *ICLR*, 2026. arXiv:2506.12362.

Jinwoo Kim, Xingyue Huang, Krzysztof Olejniczak, Kyungbin Min, Michael Bronstein, Seunghoon Hong, and İsmail İlkan Ceylan. Flock: A knowledge graph foundation model via learning on random walks. In *ICLR*, 2026. arXiv:2510.01510.

Jaejun Lee, Chanyoung Chung, and Joyce Jiyoung Whang. InGram: Inductive knowledge graph embedding via relation graphs. In *ICML*, 2023. arXiv:2305.19987.

Chaitanya Malaviya, Chandra Bhagavatula, Antoine Bosselut, and Yejin Choi. Commonsense knowledge base completion with structural and semantic context. In *AAAI*, 2020. arXiv:1910.02915.

Pouya Pezeshkpour, Yifan Tian, and Sameer Singh. Investigating robustness and interpretability of link prediction via adversarial modifications. In *NAACL*, 2019. arXiv:1905.00563.

Hongyu Ren and Jure Leskovec. Beta embeddings for multi-hop logical reasoning in knowledge graphs. In *Advances in Neural Information Processing Systems*, volume 33, pp. 19716–19726, 2020.

Tara Safavi and Danai Koutra. CoDEx: A comprehensive knowledge graph completion benchmark. In *EMNLP*, 2020. arXiv:2009.07810.

Komal Teru, Etienne Denis, and William L. Hamilton. Inductive relation prediction by subgraph reasoning. In *ICML*, 2020. arXiv:1911.06962.

Kristina Toutanova and Danqi Chen. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*, pp. 57–66, 2015.

Denny Vrandečić and Markus Krötzsch. Wikidata: A free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014. doi: 10.1145/2629489.

Jianfeng Wen, Jianxin Li, Yongyi Mao, Shini Chen, and Richong Zhang. On the representation and embedding of knowledge bases beyond binary relations. In *IJCAI*, 2016.

Yucheng Zhang, Beatrice Bevilacqua, Mikhail Galkin, and Bruno Ribeiro. TRIX: A more expressive model for zero-shot domain transfer in knowledge graphs. In *Proceedings of the Third Learning on Graphs Conference (LoG)*, volume 269 of *PMLR*, pp. 12:1–12:28, 2025. arXiv:2502.19512.

Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal Xhonneux, and Jian Tang. Neural Bellman-Ford networks: A general graph neural network framework for link prediction. In *NeurIPS*, 2021. arXiv:2106.06935.

A Proofs and Additional Statements

Proposition A.1 (Renaming)**.** *Any link-invariant KGFM is exactly (respectively, probabilistically) invariant to relation renaming T_0 with τ_{T_0} renaming the query relation accordingly.*

Proof. T_0 composed with τ_{T_0} is an isomorphism within $\mathcal{K}_{n,m}$ with $\pi = \mathrm{id}$ and ϕ the renaming. □

Proposition A.2 (Reification leaves the space)**.** *Reification of k facts maps $\mathcal{K}_{n,m} \to \mathcal{K}_{n+k,m'}$, so no isomorphism $G \simeq T_2(G)$ exists and link invariance, deterministic or probabilistic, does not constrain $f(T_2(G))$.*

Proof. Isomorphisms preserve $|V|$ and $|R|$; T_2 changes both. □

Proposition A.3 (Path expansion and reification)**.** *For any model that augments E with an inverse relation for every relation and holds no per-relation parameters, T_2 and T_3 applied to the same facts yield identical outputs.*

Proof. T_2 writes a fact as $(f, \mathrm{arg}_0, h), (f, \mathrm{arg}_1, t)$ and T_3 as $(h, r_{\mathrm{src}}, f), (f, r_{\mathrm{tgt}}, t)$. After inverse augmentation both views contain one edge in each direction between h and f and between f and t; the two augmented edge sets differ only in which label of the h–f pair is called the forward one, i.e., by a relabeling of the augmented relation set, to which the model is invariant by Prop. A.1. We verified that MRR, Hits@k and EVC agree to every reported digit at every coverage on CoDExSmall. □

Proof of Proposition 1. ULTRA augments E with (t, r^{-1}, h) for every $(h, r, t) \in E$. Under T_1 every fact (h, r, t) becomes (t, r, h), whose augmentation is (h, r^{-1}, t); the augmented multiset of $T_1(G)$ is therefore that of G with r and r^{-1} exchanged, i.e., relabeled by σ. The relation-interaction graph has one node per augmented relation and an edge of one of four types (head-to-head, head-to-tail, tail-to-head, tail-to-tail) between relations that share an entity in the corresponding positions. Exchanging r with r^{-1} exchanges heads with tails for both relations, which maps each interaction type to itself, so the interaction graph of $T_1(G)$ is σ applied to that of G with types preserved; we verified set equality (Jaccard 1.000) on CoDExSmall. The relation encoder is equivariant to this relabeling. The asymmetry is in the second stage: for a tail query $(h, r, ?)$ the relation encoder is seeded at r and the query vector is read at r; for a head query $(?, r, t)$ the model rewrites it as a tail query under r^{-1} and reads the query vector at r^{-1}, while the encoder is still seeded at r. T_1 turns every tail question into a head question and so exposes the mismatch. Seeding the encoder at the relation that is read makes both stages depend on the same relation, so the computation for $(h, r, ?)$ on G and for $(?, r, h)$ on $T_1(G)$ is identical up to σ. The measured residual (EVC 0.996–1.000 rather than exactly 1) is at the level of the floating-point non-determinism of the deterministic reference (Appendix D). □

Proof of Theorem 1. Write $H = T(G)$ and $H^- = T(G^-)$, both inverse-symmetrized, and let $N_t(h)$ be the set of nodes within t hops of h in H (the ball in H^- is contained in it). Under T, an edge e rewritten $d(e)$ times is replaced by a path of $2^{d(e)}$ edges through fresh mediators of degree two that are adjacent to nothing else, so every walk between two original entities in H projects to a walk in G with the corresponding weight and $\mathrm{dist}_H(h, v) = \ell_T(h, v)$ for original entities v; the hypothesis on E' therefore says that every edge of H derived from E' has both endpoints outside $N_L(h)$.

We compare the two executions directly. Claim: after layer $t \le L$, every node has the same state in the execution on H as in the execution on H^-, and that state is zero outside $N_t(h)$. For $t = 0$ both hold, since only the anchor is nonzero and it is the same in both executions. Suppose the claim holds for $t-1$ and consider a node v at layer t. Its aggregate is computed from the multiset of messages $\mathrm{MSG}(x_u^{(t-1)}, w_e)$ over its incident edges $e = (u, v)$. Messages from nodes u outside $N_{t-1}(h)$ are zero, because $x_u^{(t-1)} = 0$ in both executions and a message from a zero state is zero; by zero-insensitivity the aggregate depends only on the remaining multiset, whose members come from $u \in N_{t-1}(h)$. An edge with an endpoint in $N_{t-1}(h) \subseteq N_L(h)$ is not derived from E', so it is present in both H and H^-, and by the induction hypothesis $x_u^{(t-1)}$ is the same in both; the relation representation w_e is held fixed. Hence the aggregate, and therefore the updated state, coincides in the two executions. If moreover $v \notin N_t(h)$, then no neighbor of v lies in $N_{t-1}(h)$, the nonzero-message multiset is empty, the aggregate is zero, and the update of the zero state with

a zero aggregate is zero. This proves the claim, and in particular every final state and every score coincides: $s_f(H, \tau_T q) = s_f(H^-, \tau_T q)$. If f were score-invariant to T on both G and G^-, then $s_f(G, q) = s_f(H, \tau_T q) = s_f(H^-, \tau_T q) = s_f(G^-, q)$, contradicting sensitivity to E'. Finally, if a path of length ℓ from h to v has all edges rewritten d times and no path of smaller weight exists, then $\ell_T(h, v) = 2^d \ell$ by definition.

The induction uses zero-insensitivity of the aggregator to discard the messages of zero-state neighbors; a degree-normalized aggregator such as a mean does not satisfy this assumption, and whether the conclusion holds for such encoders is not settled here in either direction. The zero-message and zero-update parts of the assumption are satisfied by the layer definition of NBFNet (Zhu et al., 2021, Eq. 6) with sum aggregation once its bias and normalization terms are removed; the comparison argument is ours. □

What the theorem does not cover. The released ULTRA entity encoder applies a linear layer with bias and layer normalization after aggregation, so states outside $N_L(h)$ are not zero and depend on local degree structure; and its relation representations are computed by a second encoder over the relation-interaction graph of the whole input graph. Both let distant evidence influence a score, and we do not identify a component of the released models to which the theorem applies; it is an explanatory idealization, and the depth observations of §5 are reported as consistent with it rather than implied by it. The proof assumes a zero-insensitive aggregator (sum qualifies; a mean does not satisfy the assumption) and excludes bias and normalization terms.

Why no composition bound is stated. If f were exactly invariant to each rewrite T_i on the whole orbit $\{T_{i_k} \circ \cdots \circ T_{i_1}(G)\}$, it would be invariant to every finite composition, and an approximate bound $d(f(G), f(T_i G)) \leq \epsilon_i$ holding uniformly on the orbit would give $\sum_j \epsilon_{i_j}$ by the triangle inequality. The uniformity hypothesis is the whole content of such a statement, and we only measure ϵ_i on original graphs, so we do not state it as a result.

Proof of Proposition 2. (i) Let (a, b) be a key that T created. By hypothesis, every edge labeled a or b is incident to a mediator that T created, and every such mediator has the motif's degree signature, so it is a candidate for the key (a, b); the mediator-side endpoint of every a- or b-edge is therefore a candidate of that family, giving $\pi_a = \pi_b = 1 \geq \tau$. The key has support at least s by the stated proviso. Coverage does not enter, because unrewritten facts keep their original relation, which is neither a nor b. (ii) This is the acceptance rule read contrapositively: acceptance requires support at least s and $\pi_a \geq \tau$, $\pi_b \geq \tau$, where π_r counts every edge labeled r in the graph. On an untouched graph, all relations are genuine, so a key is accepted, and its candidates are collapsed exactly when the stated confinement holds. (iii) A pass collapses every candidate of every accepted key by removing all edges incident to the candidate set D and adding, for each $v \in D$, one edge between v's two former neighbors. By the pairwise non-adjacency assumption, no neighbor of a collapsed node is itself in D, so every recovered edge joins two surviving nodes, no node of D remains as an endpoint, and every node of D leaves the graph; no node is added. Hence, a non-empty pass strictly decreases $|V|$ and the unbounded iteration ends within $|V|$ passes. Without non-adjacency, this step fails: two adjacent collapsed nodes each survive as an endpoint of the other's recovered edge. For $T^d(G)$ under the stated conditions, consider a mediator f created at level $j < d$: both of its role edges were rewritten at level $j + 1$, so in $T^d(G)$ it has in-degree 2 and out-degree 0 and carries neither motif signature; only level-d mediators are candidates. Their keys are created keys, accepted by (i), and no genuine key is accepted by assumption, so the first pass collapses exactly the level-d mediators and restores the role edges of level $d - 1$ under recovered relations that occur only on level-$(d - 1)$ mediators. Induction on d gives one level per pass and the original graph after d collapsing passes; the following pass collapses nothing, and the loop stops. □

The proof of (iii) uses $\tau \leq 1$ through (i) only; the experiments use $\tau = 0.9$ and $s = 2$, and the purity-filtered runs of Appendix B terminated after $d + 1$ passes on every rewritten graph we tested. A support threshold of $s = 2$ leaves a created key unaccepted when a relation was rewritten in a single fact; this is the single-edge-mediator residual reported in §6.

B THE STRUCTURAL NORMALIZER

Algorithm 1 gives the procedure used in §6. We describe it in five parts.

(a) Candidate motifs. A mediator introduced by a rewrite has a characteristic degree signature: a *reification* mediator has no incoming edges and exactly two outgoing edges under two distinct relations $a < b$, and a *path* mediator has exactly one incoming edge under a and one outgoing edge under b. Every node with such a signature is a candidate, whatever its origin, and is tagged with its *key*, the motif family together with the relation pair. Candidate detection uses graph structure only; it does not use a rewrite manifest or labels identifying introduced mediator nodes.

(b) Support and relation purity. A key is a plausible encoding convention only if it recurs: it is accepted when at least s distinct nodes carry it ($s = 2$ throughout, $s = 1$ where stated). Recurrence alone accepts genuine structures on graphs that contain mediator-shaped entities (§6). The *relation-purity* test removes such keys by a property that every created relation has, and a genuine relation typically lacks: a relation created by a rewrite occurs only on the mediators it created. For each motif family, let π_r be the fraction of all edges labeled r whose mediator-side endpoint is a candidate of some accepted key of that family; an accepted key (a, b) is kept only if $\min(\pi_a, \pi_b) \geq \tau$ ($\tau = 0.9$ throughout; Fig. 10 shows that any $\tau \geq 0.7$ gives the same counts on the graphs tested). Purity is measured per family because the two motifs use their relations in opposite roles, so a relation can be pure for one family and not the other. What the test cannot decide is a relation with a single edge: its one mediator has support 1 and purity trivially 1. With $s = 2$ such mediators are left in place (74 of 193,950 on AristoV4, 27 of 478,429 on DBpedia100k), and no genuine node was collapsed on any tested graph; with $s = 1$ they are recovered but a few genuine single-edge nodes are collapsed instead (5 on DBpedia100k, none on the other graphs tested). No structural rule separates a single-edge mediator from a single-edge entity, so both settings are reported.

(c) Simultaneous collapse and relation mapping. Within a pass, the set D of nodes with an accepted key is fixed before any edit; every edge incident to D is removed at once; and for each collapsed node, one recovered edge is added between the two neighbors it had before removal, labeled with a fresh relation that is specific to the key, so that facts encoded under different keys remain distinguishable. Therefore, processing order does not matter. A node with no remaining edges leaves the graph; this is how mediators are removed. Two collapsed nodes that were adjacent are the exception: each remains as an endpoint of the other's recovered edge, and a recovered edge can give a surviving node a motif signature in the next pass. A candidate whose two edges end at the same entity, which a rewrite produces only from a self-loop fact, is collapsed to a recovered self-loop and thus restores that fact. The output is the normalized graph together with the mapping from surviving and recovered relations to their identifiers. Evaluation then proceeds on the original candidate entities, which are retained, and on the query relations, which are mapped via this relation mapping; this bookkeeping is part of the evaluation procedure and plays no role in candidate detection. Because the output identifies surviving mediators only by their positions relative to the original entities, the identities of individual surviving mediators are not exposed. The *partial-recognition* experiments withhold a random half of the accepted keys (after the purity test), so that every instance of a withheld key stays reified; they model a normalizer whose motif library is incomplete.

(d) Iteration and stopping. Composed rewrites nest mediators, and a mediator created at an inner level carries no motif signature until the outer level has been collapsed (Prop. 2), so the pass is repeated. The iteration stops when a pass collapses no node; when the optional yield rule is enabled, it also stops when a pass collapses fewer than 5% of the nodes of the previous pass, and that pass is discarded; and in every case, it stops after at most six passes. This execution policy is distinct from Prop. 2(iii), which guarantees termination of the unbounded iteration under the pairwise non-adjacency assumption. Three outcomes were observed. On the support-only variant, iteration on the untouched DBpedia100k graph and on its depth-1 and depth-2 views reached the six-pass cap without converging: after the first passes, each further pass collapsed 14 nodes with the triple count unchanged (whether graph states recur was not checked); the WN18RR depth-1 view showed the same pattern with 18 nodes; on CoDExSmall the second pass collapsed nothing and the iteration converged. Every support-only "iterated" result on DBpedia100k and WN18RR (Tables 15, 16; Figs. 3a, 9) is therefore a capped result and the CoDExSmall one a converged one. All purity-filtered

iterations stopped on their own after d collapsing passes plus one final pass that collapsed nothing. The five-fold normalizer runs, the untouched-graph runs, and Figs. 3b,c apply a single pass. Pass counts were recorded for the fold-1 runs; the fold-2 rows of Table 16 are labeled by analogy.

(e) Cost and limitations. One pass costs time linear in the number of edges plus the size of the degree tables; an iterated run costs at most six such passes on the current graph. Measured on CPU (Table 23), view generation takes seconds to two minutes, one normalizer pass at most 17 s on the largest reified graph and 20 s with the purity test, and node inflation under context reification is 2–35×; iterated runs were not timed separately. Three levels of verification are distinguished. An edge-level round trip, equality of the edge partition by relation up to relation renaming, is verified on a small graph with $s = 1$. On benchmark graphs only the edge and relation counts of the normalized graph were compared with the original's; they agree for CoDExSmall, WN18RR and DBpedia100k at depths 1 and 2 and for AristoV4 with $s = 1$, which is a count check, not an edge-level comparison. Scored runs report MRR at three or four decimals and top-1 agreement. "Near-original MRR" means a difference within the stated tolerance at the displayed precision, and "equal at reported precision" means the displayed digits coincide; an EVC of 1.00 describes top-1 agreement at the displayed precision and at the level of the deterministic reference, and establishes neither identical rankings, nor identical scores, nor identical MRR (e.g., AristoV4 at support 1: EVC 1.00 with MRR 0.508 vs 0.506, Table 15). A normalizer whose library contains only the reification motif recovers nothing of the Freebase-style expansion, which has the path shape (in a preliminary run, MRR $0.3036 \rightarrow 0.3040$ with EVC 0.408). The oracle-boundary variant used for comparison receives the labels identifying introduced mediators and never considers an original entity.

Algorithm 1: Blind structural normalization with the relation-purity test

Input: graph $G = (V, R, E)$; support threshold s (default 2); purity threshold τ (default 0.9); pass limit (6)
Output: normalized graph and the mapping of surviving and recovered relations

repeat
  tabulate the in- and out-edges of every node;
  foreach *node* v **do**
    if $\mathrm{in}(v) = 0$ *and* $\mathrm{out}(v) = 2$ *with distinct relations* $a < b$ **then** $\mathrm{key}(v) \leftarrow (\mathrm{R}, a, b)$;
    if $\mathrm{in}(v) = 1$ *under* a *and* $\mathrm{out}(v) = 1$ *under* b **then** $\mathrm{key}(v) \leftarrow (\mathrm{P}, a, b)$;
  end
  accept every key carried by at least s distinct nodes;
  foreach *motif family* $F \in \{\mathrm{R}, \mathrm{P}\}$ **do**
    $C_F \leftarrow$ union of candidates of all accepted keys of family F;
    foreach *relation* r **do**
      $\pi_r \leftarrow$ fraction of edges labeled r whose mediator-side endpoint is in C_F
    end
    reject every accepted key (F, a, b) with $\min(\pi_a, \pi_b) < \tau$;
  end
  $D \leftarrow$ set of all nodes with an accepted key (fixed for this pass);
  remove every edge with an endpoint in D;
  foreach *node* $v \in D$ *with neighbors* x, y *recorded before removal* **do**
    add $(x, \rho_{(F,a,b)}, y)$ with a fresh relation $\rho_{(F,a,b)}$ per key;
  end
  renumber relations (surviving first, then recovered) and nodes (original entities keep their identifiers; surviving nodes follow);
until *no node was collapsed in the last pass, or the pass limit is reached*;
return the graph and the relation mapping;

C Experimental Details

C.1 Background on the backbones

This summary is meant to make the paper self-contained; the cited papers give the full definitions. *ULTRA* (Galkin et al., 2024a) answers a query $(h, r, ?)$ in two stages. First, it builds a *relation graph* whose nodes are the relations of the input graph (with their inverses) and whose edges record four ways two relations can share an entity (head-to-head, head-to-tail, tail-to-head, tail-to-tail). A six-layer NBFNet (Zhu et al., 2021) over this relation graph, seeded at the query relation r, produces a representation for every relation. Second, a six-layer conditional message-passing network over the entity graph, seeded at the anchor entity h and using those relation representations as edge features, produces a state for every entity, which is scored as a candidate answer. Head queries $(?, r, t)$ are answered as tail queries under r^{-1}; Prop. 1 concerns which relation seeds the first stage in that case. The checkpoint `ultra_3g` is pretrained on FB15k-237, WN18RR, and CoDEx-Medium. *TRIX* (Zhang et al., 2025) keeps the same two ingredients but interleaves relation and entity updates across layers. *MOTIF* (Huang et al., 2025) replaces the relation graph by a relation *hypergraph* whose hyperedges are small relational motifs, encoded by a hypergraph network; its relation encoder is not seeded at the query relation, which is why it is inversion-invariant as shipped. *FLOCK* (Kim et al., 2026) does not use message passing: it samples anonymized random walks (entities and relations replaced by their first-occurrence indices), encodes each walk with a recurrent network, refines entity states over several rounds, and averages over independent walk samples; its predictions are therefore stochastic, which is why its consistency reference is measured by re-scoring the original graph. *UltraQuery* (Galkin et al., 2024b) composes ULTRA projections with fuzzy logical operators to answer conjunctive, disjunctive, and negated queries.

C.2 Datasets

Table 4 lists the graphs. All are taken from ULTRA's dataset collection with their standard splits. CoDExSmall, DBpedia100k, AristoV4, Hetionet, ConceptNet100k, and YAGO3-10 are outside every backbone's pretraining mixture. FB15k-237 and WN18RR are inside the pretraining mixture of ULTRA (`ultra_3g`: FB15k-237, WN18RR, CoDEx-Medium), MOTIF (same three graphs), TRIX (same three graphs) and FLOCK (same three graphs), so the results there are transductive rather than zero-shot.

Table 4: Knowledge graphs used in this paper. $|V|$, $|R|$ and $|E|$ count entities, relation types and training edges; "triples / fold" is the seeded per-fold cap on test triples for the ULTRA diagnosis, each triple yielding two directional queries (so 600 triples over five folds give the 6,000 queries of Table 2; "full split" means no cap); the last column states whether the graph is in the backbones' pretraining mixture.

Graph	$\lvert V\rvert$	$\lvert R\rvert$	$\lvert E\rvert$	triples / fold	pretraining
CoDExSmall	2,034	42	32,888	full split	no
WN18RR	40,943	11	86,835	600	yes (all backbones)
ConceptNet100k	78,334	34	100,000	full split	no
AristoV4	44,949	1,605	242,567	600	no
FB15k-237	14,541	237	272,115	600	yes (all backbones)
DBpedia100k	99,604	470	597,572	1,000	no
YAGO3-10	123,182	37	1,079,040	safety only	no
Hetionet	45,158	24	2,025,177	500	no

C.3 Protocol, seeds and uncertainty

Relations are partitioned into $K = 5$ folds by a deterministic greedy assignment that balances the folds by test frequency; the partition does not depend on the seed. Each test triple whose relation is in the fold yields two directional queries (head and tail prediction), scored over the semantic entity range with the filtered protocol; the filtered set is mapped from the original graph. Where a per-fold cap is used (Table 4), a seeded subsample of test triples is drawn. *What n counts.* In Table 1 and in

the FLOCK comparison, n is the number of directional queries pooled over the five folds for one seed (e.g., CoDExSmall: 1,828 test triples $\times 2$ directions $= 3{,}656$; the FLOCK pairs use per-fold caps of 80 or 150 triples, giving 800 or 1,500 queries); in Tables 4 and 6, n is the per-fold sampling cap on test triples; fold-level results, such as the CoDExSmall fold-1 original MRR of 0.799 in Tables 15 and 14, cover a single fold's queries and differ from the pooled value because folds contain different relations. *Seeds.* Two seeds, 0 and 1, are used. The seed fixes the test subsample, the fresh-node identifiers, the mixed-schema slice assignment, the matched-noise wiring, the negative samples, and the initialization of fine-tuning; each run is identified by its seed. For CoDExSmall and ConceptNet100k, the cap covers the whole test split, so a second seed changes only the fresh-node identifiers, to which the models are equivariant, and the noise wiring; the reification rows therefore coincide, and only DBpedia100k, WN18RR, AristoV4, Hetionet, and FB15k-237 are replications over different query subsamples. *Uncertainty.* Intervals written $[\cdot, \cdot]$ are 95% paired bootstrap intervals over queries (10,000 resamples, seed 0), which quantify query-resampling uncertainty for one run; we use the bootstrap because per-query reciprocal ranks are bounded and strongly skewed, and paired because every view is scored on the same queries. Values written $\pm$ are means and standard deviations across fold/seed runs (Table 13 and §6.1) and quantify variability across folds and training seeds, which is a different source. The schema cost ratio is computed per run from pooled MRRs; where two seeds are averaged, the reported SCR is the mean of the two per-seed ratios. The deletion precondition is met when the paired bootstrap interval of the deletion effect excludes zero in the negative direction.

C.4 Complex queries

We use the FB15k-237 logical-query benchmark (the BetaE splits of Ren & Leskovec (2020), 14 query types) with UltraQuery's released checkpoint and its own filtered evaluation. Base relations are split into $K = 2$ folds; for fold k we evaluate only test queries whose *every* projection uses a fold-k relation and rewrite all relations outside the fold, so the rewrite is answer-preserving by construction. Predictions are restricted to the original candidate entities. Two folds and two partition seeds with 400 queries per type per run give 22,184 evaluations.

C.5 Learned remedies

Let $s^{(v)} \in \mathbb{R}^{B\times(1+K)}$ be the score matrix of a batch on view v over one positive and $K = 32$ negatives, the same candidates in every view, and let $p^{(v)} = \mathrm{softmax}(s^{(v)})$. With $\mathcal{L}_{\mathrm{qa}}$ denoting ULTRA's adversarially weighted binary cross-entropy (Galkin et al., 2024a), JS the Jensen–Shannon divergence between rows, o the original view, e an answer-preserving rewritten view, and c a meaning-changing view (the same mediators rewired at random), the three objectives are

$$\mathcal{L}_{\mathrm{aug}} = \mathcal{L}_{\mathrm{qa}}(s^{(o)}) + \mathcal{L}_{\mathrm{qa}}(s^{(e)}), \tag{1}$$

$$\mathcal{L}_{\mathrm{pc}} = \mathcal{L}_{\mathrm{aug}} + \mathrm{JS}\big(p^{(o)}, p^{(e)}\big), \tag{2}$$

$$\mathcal{L}_{\mathrm{pc+neg}} = \mathcal{L}_{\mathrm{pc}} + \max\big(0,\ \gamma + \mathrm{JS}(p^{(o)}, p^{(e)}) - \mathrm{JS}(p^{(o)}, p^{(c)})\big), \tag{3}$$

with margin $\gamma = 0.1$ and softmax temperature 1. Every active term has weight 1. All fine-tuning starts from `ultra_3g` and uses Adam with learning rate 2×10^{-4}, batch size 8 (2 on DBpedia100k), and negatives sampled once from the semantic entity range and reused verbatim in every view. Held-out-family runs (Table 13) use 1,500 steps on CoDExSmall and 800 on DBpedia100k; composition runs (Table 15) use 1,200 steps; view-specific fine-tuning (Table 14) uses 2,000 steps at learning rate 5×10^{-4} on CoDExSmall and 1,500 steps on DBpedia100k; the conditioning fine-tuning (Table 10) uses 2,000 steps on CoDExSmall and 1,500 on DBpedia100k. *Tuning.* No hyperparameter was tuned on a test view. Learning rates and step counts were taken from ULTRA's own fine-tuning configuration and not searched; the loss weights and margin were fixed before the first run. The only swept quantities are the purity threshold τ (eleven values in $[0, 1]$, Fig. 10, chosen for safety on untouched graphs, not for accuracy) and the support threshold $s \in \{1, 2\}$, for which both settings are reported.

C.6 FLOCK budget and the consistency reference

FLOCK is run from its released entity-prediction checkpoint with 16 walks and 16 ensemble samples on CoDExSmall, and 8 ensemble passes on the larger graphs, with a per-fold cap of 80 test triples

(150 on CoDExSmall), each yielding two directional queries, i.e., 800 (1,500) queries over the five folds, paired exactly with an ULTRA run on the same queries. Its consistency reference is the top-1 agreement between two re-scorings of the same original graph with different seeds, listed per block in Table 11. This reference is an empirical resampling level, not a ceiling: NCD can be negative when a rewritten view happens to agree with the first scoring more often than a second scoring does, and an NCD near zero indicates agreement at the resampling level, not equality of the score distributions. This budget is below the one used in Kim et al. (2026) for the large graphs, which is why FLOCK's absolute MRR on DBpedia100k, AristoV4 and Hetionet is below its published value (Table 5); the comparison in §5.4 is about each model's agreement loss relative to its own reference, not about absolute accuracy, and it is limited to the tested budget.

C.7 Hardware and software

All runs use one node with four NVIDIA Quadro RTX 8000 GPUs (48 GB each, compute capability sm_75, driver 580.173), 32 CPU cores, and 251 GB of RAM, running Linux with Python 3.12, PyTorch 2.4.0 (CUDA 12.1), PyTorch Geometric 2.6.1, torch-scatter 2.1.2, and torch-sparse 0.6.18, all in fp32. The fused sparse kernel of the reference implementations was compiled with CUDA 12.1 and gcc 12.4 (Appendix D explains why it is replaced at runtime). The diagnosis of one graph takes minutes (CoDExSmall) to about 15 hours (AristoV4 with FLOCK); a fine-tuning run takes under two hours. The normalizer and view generation run on CPU (Table 23).

D Reproduction and Evaluation Checks

D.1 Pretrained models and reproduction of published numbers

All ULTRA results use the checkpoint pretrained on FB15k-237, WN18RR and CoDEx-Medium (`ultra_3g`); the checkpoint that adds NELL-995 to the mixture does not reproduce the published zero-shot numbers (CoDExSmall 0.464 vs published 0.472) and is not used. MOTIF and TRIX use their released checkpoints with the same three-graph pretraining, and FLOCK its released entity-prediction checkpoint at the reduced walk budget of Appendix C. Table 5 compares our original-view accuracy with the zero-shot entity-prediction numbers published by Kim et al. (2026) (their Table 9), which is also the source of the published ULTRA and TRIX columns. ULTRA and TRIX reproduce their published values within 0.002 on CoDExSmall, DBpedia100k and AristoV4. Two discrepancies remain. ConceptNet100k does not reproduce with either ULTRA checkpoint on the standard splits (0.061 and 0.125 against a published 0.082), and we could not identify the cause; its MRR values are reported (Table 1), but its absolute accuracy is not comparable to the published number, and its schema cost ratio is withheld because deletion helps. FLOCK's absolute accuracy on DBpedia100k, AristoV4, and Hetionet is below its published value because of the reduced walk and ensemble budget; the FLOCK comparison in §5.4 therefore concerns each model's agreement loss relative to its own reference, not absolute accuracy.

D.2 Evaluation safeguards

Three checks are applied to every scored run, each motivated by a failure that would otherwise pass silently. *Message-passing direction.* The reference implementations' non-fused message-passing path can aggregate in the transposed direction on the software configuration used here, which changes the score of every query without raising an error. We therefore evaluate the sparse relational message-passing operation through an exact chunked re-implementation whose outputs are verified against the fused reference kernel on the graphs where that kernel runs, and we require the original-view MRR to match the published value before a graph enters the study (Table 5). *Non-finite predictions.* A candidate row with non-finite scores would be ranked as if the reference answer were first, inflating MRR while lowering agreement. Every result in this paper is computed with non-finite score rows excluded and their count recorded; in the released results, no such row occurs in any run reported. *Pairing.* Cross-view consistency metrics pair predictions by the semantic question (anchor entity, relation, direction) rather than by evaluation position, because global inversion exchanges head and tail queries; pairing by position would compare different questions and report near-zero agreement under T_1 for reasons unrelated to the model. These checks validate the evaluation procedure used; they do not establish the correctness of every component of the reference implementations.

Table 5: Original-view MRR of our evaluation procedure against published zero-shot MRR from Kim et al. (2026). Our ULTRA and TRIX rows use the full context-only protocol on the largest evaluation cap; our FLOCK rows use the reduced walk budget of Appendix C on the matched subsample (800 directional queries, 1,500 on CoDExSmall). Published FLOCK numbers use a larger walk and ensemble budget. †: reproduction check on the full CoDExSmall test split (16 walks, 16 samples), from a preliminary run not part of the released result set; the matched 1,500-query subsamples used in Table 11 give original-view MRR 0.456 (seed 0) and 0.480 (seed 1). WN18RR is not zero-shot for any backbone and is listed for completeness.

	ULTRA		TRIX		FLOCK	
Graph	published	ours	published	ours	published	ours
CoDExSmall	0.472	0.472	0.472	0.473	0.441	0.439†
DBpedia100k	0.398	0.396	0.426	0.428	0.450	0.310–0.324
AristoV4	0.182	0.183	0.181	0.183	0.308	0.146–0.166
Hetionet	0.257	0.244	0.279	–	0.246	0.094
ConceptNet100k	0.082	0.061	0.193	–	0.248	–
WN18RR	–	0.375	–	0.502	–	0.419

D.3 Consistency reference

Scoring the same graph with the same deterministic model twice gives EVC 0.9987, because the floating-point order of parallel aggregation is not fixed and a few near-ties flip. This is the reference for the deterministic models; an inversion result described as "at the level of the deterministic reference" (EVC 0.996–1.000) is agreement at that level, not verified equality of unrounded scores. The reference for the stochastic model is its own agreement between two re-scorings (Appendix C).

E Additional Results

This appendix contains every table and figure referenced in the main text but not shown there, in the order of the main text.

E.1 Diagnosis

Table 6: Full version of Table 3: three message-passing KGFMs under context-only reification. n here is the per-fold sampling cap on test triples (each yielding two directional queries), not the pooled directional-query count of Table 1; for CoDExSmall, the cap of 1,200 exceeds every fold, so all 1,828 test triples (3,656 queries) are evaluated. WN18RR is in each model's pretraining mixture. †: deletion helps this backbone on this graph, so SCR is withheld. MOTIF's DBpedia100k row uses a smaller cap.

Dataset	backbone	n	orig. MRR	reified	Δ	EVC	inv. EVC
AristoV4	ULTRA†	600	0.185	0.151	+18.0%	0.097	0.282
AristoV4	TRIX†	600	0.183	0.216	-18.2%	0.370	0.388
CoDExSmall	ULTRA	1200	0.472	0.355	+24.6%	0.389	0.437
CoDExSmall	MOTIF	1200	0.474	0.411	+13.4%	0.284	1.000
CoDExSmall	TRIX	1200	0.473	0.436	+7.8%	0.450	0.364
DBpedia100k	ULTRA	1000	0.399	0.182	+54.3%	0.251	0.467
DBpedia100k	MOTIF	200	0.378	0.279	+26.2%	0.371	0.998
DBpedia100k	TRIX	1000	0.428	0.351	+18.0%	0.456	0.488
WN18RR	ULTRA	600	0.374	0.241	+35.7%	0.218	0.267
WN18RR	MOTIF	600	0.513	0.387	+24.6%	0.298	0.999
WN18RR	TRIX	600	0.502	0.408	+18.6%	0.473	0.601

Table 7: Freebase-style expansion of FB15k-237, both seeds, with all normalizer variants (path-aware without purity test; with purity test; with half of the keys recognized). $n = 6{,}000$ directional queries per seed; transductive. Main-text Table 2 shows the subset with the purity test.

View	MRR	H@1	H@10	EVC	Δ
Original (collapsed, as shipped)	0.365	0.267	0.559	–	–
Freebase-style expansion	0.304	0.224	0.455	0.408	+16.8%
+ blind normalizer (path-aware)	0.365	0.268	0.559	0.946	-0.1%
+ blind normalizer, purity test	0.365	0.267	0.559	1.000	+0.0%
+ normalizer, 50% keys recognized	0.332	0.246	0.503	0.535	+9.0%
+ normalizer, purity test, 50% keys	0.338	0.251	0.509	0.561	+7.4%
Synthetic reification, same relations	0.296	0.216	0.446	0.398	+19.0%
Control: relations deleted	0.345	0.256	0.523	0.472	+5.4%
Control: matched-structure noise	0.225	0.142	0.394	0.229	+38.4%
Original (collapsed, as shipped) (seed 1)	0.363	0.263	0.564	–	–
Freebase-style expansion (seed 1)	0.299	0.218	0.456	0.402	+17.8%
+ blind normalizer, purity test (seed 1)	0.363	0.263	0.564	1.000	+0.0%
Synthetic reification, same relations (seed 1)	0.290	0.211	0.443	0.388	+20.1%
Control: relations deleted (seed 1)	0.346	0.254	0.532	0.480	+4.8%
Control: matched-structure noise (seed 1)	0.227	0.148	0.396	0.247	+37.6%

Table 8: Context-only reification under complex logical queries (UltraQuery on FB15k-237, 22,184 query evaluations pooled over two relation folds × two partition seeds; "Control" rows are the deletion and matched-noise reference conditions). EVC by query type; UltraQuery runs the entity encoder once per projection.

View	MRR	H@10	Δ MRR	EVC
original	0.172	0.296	–	–
T_2 **reified**	0.143	0.242	+16.8%	0.248
Control: relations deleted	0.165	0.275	+4.3%	0.398
Control: matched-structure noise	0.107	0.187	+38.1%	0.197
EVC *under reification by query type*				
1p: 0.42 2p: 0.18 3p: 0.18 2i: 0.23 3i: 0.49				
ip: 0.19 pi: 0.34 2u-DNF: 0.42 up-DNF: 0.18 2in: 0.14				
3in: 0.27 inp: 0.16 pin: 0.11 pni: 0.15				

Table 9: Global inversion at inference time: the released ULTRA checkpoint scored under the shipped and under the consistent conditioning of Prop. 1 ($n = 800$ queries per graph, seed 0). The last column is the change in original-view MRR under the consistent conditioning before any fine-tuning. A tied-interaction alternative that we first hypothesized does not restore invariance on CoDExSmall (EVC 0.216) and costs 24% relative MRR.

Dataset	as shipped			consistent conditioning			cost on orig.
	orig	inverted	EVC	orig	inverted	EVC	
AristoV4	0.310	0.272	0.211	0.280	0.280	0.999	-9.8%
CoDExSmall	0.471	0.440	0.246	0.456	0.456	1.000	-3.3%
DBpedia100k	0.396	0.383	0.476	0.416	0.416	0.999	+5.0%
Hetionet	0.298	0.214	0.127	0.191	0.191	1.000	-35.8%

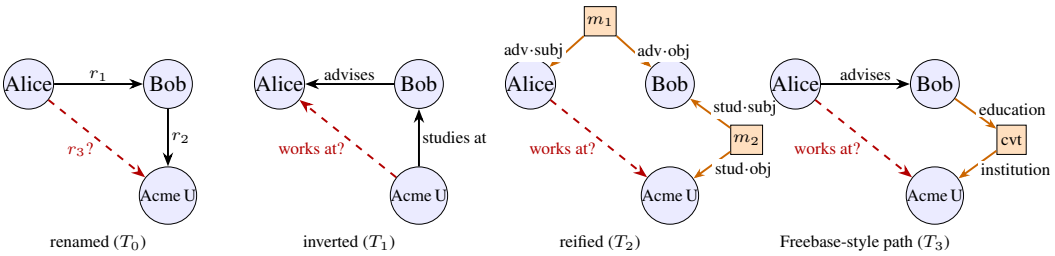


Figure 4: The toy context of Fig. 1 under four encodings. Renaming (T_0) renames every relation, including the query relation through τ_{T_0}, and global inversion (T_1) reverses every fact and translates the query direction; both keep one hop per fact. Reification (T_2) and the Freebase-style path (T_3) place a mediator on each rewritten fact. Measured top-1 agreement for ULTRA on CoDExSmall: 1.00 under T_0 and 0.39 under T_2 (0.99 after the purity-filtered normalizer; Tables 1, 17, five folds pooled); under global inversion 0.44 in the five-fold evaluation of Table 3 and 0.25 in the 800-query inversion experiment of Table 9, where the consistent conditioning gives 1.00; for the Freebase-style expansion of FB15k-237, 0.41 (1.00 after normalization; Table 2).

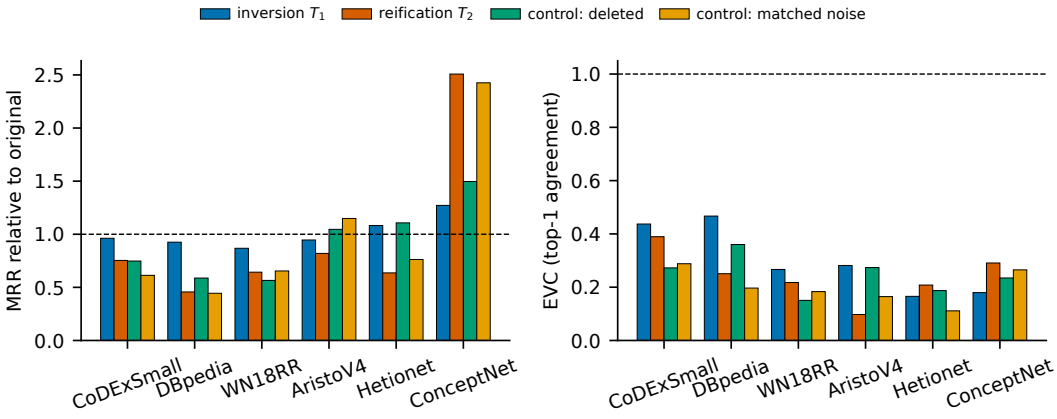


Figure 5: Reification against its reference conditions (ULTRA, `ultra_3g`, five-fold pooled, seed 0; the deletion and matched-noise conditions are controls, not lower bounds). Left: MRR relative to the original view; right: top-1 agreement. Where deletion measurably hurts, reification lands near the deletion condition (above it on WN18RR, SCR 0.82; below it on DBpedia100k, 1.26–1.32), and the matched-noise condition shows that the structure of the same node and edge count and mediator degree does not account for the loss, without controlling every topological difference. Agreement is 0.10–0.39 on every graph against a renaming reference of 0.999. On ConceptNet100k, AristoV4, and Hetionet deletion helps; MRR still measures accuracy there, and only the schema cost ratio is withheld.

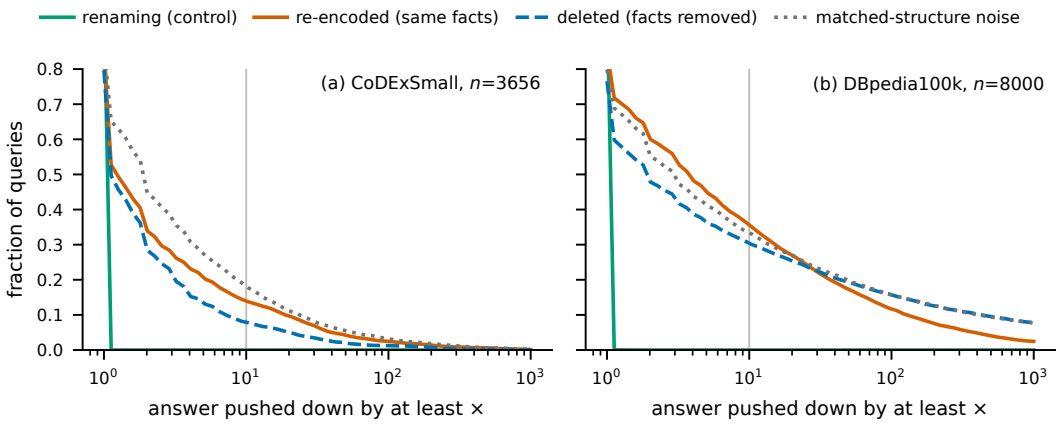


Figure 6: Per-query rank displacement (ULTRA, five folds pooled, both directions, seed 0). Fraction of queries whose reference answer fell by at least the factor on the horizontal axis: (a) CoDExSmall, (b) DBpedia100k. Renaming is flat at zero, as Prop. A.1 requires. On CoDExSmall, re-encoding and deleting cost the same MRR (SCR 0.98), yet re-encoding pushes 13.8% of answers more than tenfold down the ranking against deletion's 7.7%; on DBpedia100k, re-encoding is worse on both counts (73.0% vs 61.0% displaced; 35.4% vs 30.3% beyond tenfold).

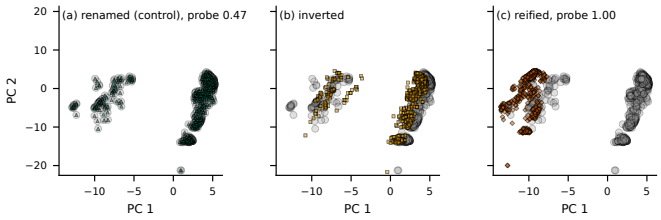


Figure 7: ULTRA's final candidate states projected to two dimensions (CoDExSmall, fold 1, seeded PCA), each rewrite (color) against the original (gray, drawn larger beneath). (a) Renaming leaves the states in place, and a linear probe cannot separate the two views (accuracy 0.47, chance). (b) Inversion moves them partway. (c) Reification moves them into a separate region, and the same probe separates the views with accuracy 1.000. The encoding is therefore linearly decodable from the state; this does not by itself imply that an invariant model would have to discard it.

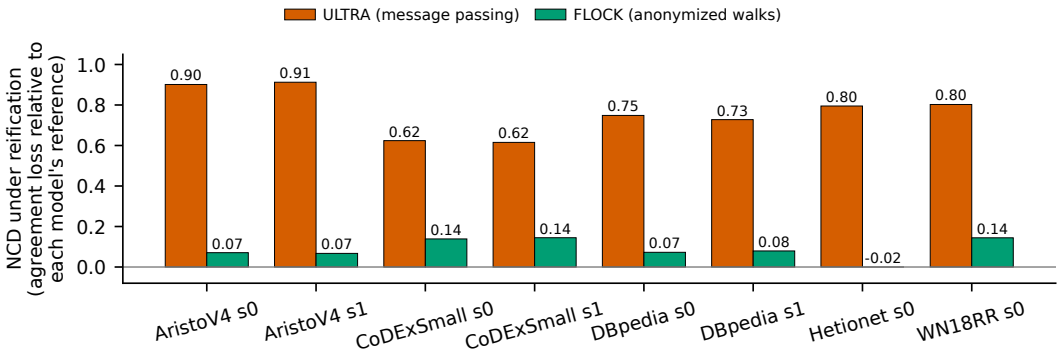


Figure 8: Normalized consistency deficit under reification (agreement loss relative to each model's own reference, §3) for ULTRA and FLOCK on identical queries with matched three-graph pretraining (five graphs; two seeds for three of them). FLOCK loses 0.07–0.14 of its reference agreement where ULTRA loses 0.62–0.91; on Hetionet FLOCK's value is −0.02, within resampling noise. The comparison is at FLOCK's reduced walk budget, and its absolute MRR on Hetionet is well below ULTRA's (Table 11).

Table 10: Global inversion after matched fine-tuning on the original view under each conditioning (CoDExSmall folds 1–4 and DBpedia100k folds 1–2, seed 0). Under the consistent conditioning EVC is 0.996–1.000 on every run; the original-view delta, with its 95% paired bootstrap interval, has no consistent sign (mean -0.001, sd 0.017 over six runs).

Dataset, fold	conditioning	original	inverted	$\mathrm{EVC}_{\mathrm{inv}}$	reified
CoDExSmall, fold 1 (s0)	as shipped	0.827	0.821	0.744	0.754
CoDExSmall, fold 1 (s0)	consistent	0.836	0.836	1.000	0.817
original-view δ of the consistent conditioning: +0.0098 [+0.0025, +0.0175]					
CoDExSmall, fold 2 (s0)	as shipped	0.605	0.578	0.282	0.550
CoDExSmall, fold 2 (s0)	consistent	0.600	0.600	1.000	0.541
original-view δ of the consistent conditioning: -0.0048 [-0.0165, +0.0068]					
CoDExSmall, fold 3 (s0)	as shipped	0.462	0.402	0.477	0.323
CoDExSmall, fold 3 (s0)	consistent	0.463	0.463	1.000	0.337
original-view δ of the consistent conditioning: +0.0006 [-0.0121, +0.0134]					
CoDExSmall, fold 4 (s0)	as shipped	0.417	0.386	0.451	0.338
CoDExSmall, fold 4 (s0)	consistent	0.412	0.412	1.000	0.316
original-view δ of the consistent conditioning: -0.0050 [-0.0174, +0.0074]					
DBpedia100k, fold 1 (s0)	as shipped	0.595	0.464	0.521	0.323
DBpedia100k, fold 1 (s0)	consistent	0.567	0.567	0.996	0.396
original-view δ of the consistent conditioning: -0.0281 [-0.0409, -0.0151]					
DBpedia100k, fold 2 (s0)	as shipped	0.463	0.430	0.564	0.285
DBpedia100k, fold 2 (s0)	consistent	0.486	0.485	1.000	0.233
original-view δ of the consistent conditioning: +0.0229 [+0.0074, +0.0385]					

Table 11: ULTRA versus FLOCK on identical query subsamples with matched three-graph pretraining (five graphs, eight dataset/seed pairs). EVC is raw top-1 agreement with the original view; NCD normalizes it by each model's own reference, listed per block together with n and FLOCK's walk budget. "control: deleted" is the deletion reference condition.

Dataset	view	ULTRA				FLOCK			
		MRR	Δ	EVC	NCD	MRR	Δ	EVC	NCD
AristoV4 s0	T1 inversion	0.202	+2.4%	0.289	0.71	0.166	-3.7%	0.691	-0.00
AristoV4 s0	control: deleted	0.215	-3.7%	0.243	0.76	0.230	-43.8%	0.345	0.50
AristoV4 s0	T2 reified	0.173	+16.4%	0.099	0.90	0.149	+7.1%	0.640	0.07
reference EVC: ULTRA 1.000 (renaming), FLOCK 0.689 (re-scored original); n = 800 queries; FLOCK: 16 walks, 8 samples									
AristoV4 s1	T1 inversion	0.171	+6.4%	0.287	0.71	0.146	-4.6%	0.704	0.00
AristoV4 s1	control: deleted	0.182	+0.6%	0.266	0.73	0.184	-31.5%	0.386	0.45
AristoV4 s1	T2 reified	0.133	+27.3%	0.087	0.91	0.119	+14.6%	0.659	0.07
reference EVC: ULTRA 0.999 (renaming), FLOCK 0.706 (re-scored original); n = 800 queries; FLOCK: 16 walks, 8 samples									
CoDExSmall s0	T1 inversion	0.477	+4.6%	0.464	0.54	0.469	-2.8%	0.481	-0.01
CoDExSmall s0	control: deleted	0.364	+27.2%	0.325	0.68	0.441	+3.3%	0.275	0.42
CoDExSmall s0	T2 reified	0.361	+27.9%	0.376	0.62	0.401	+12.0%	0.410	0.14
reference EVC: ULTRA 1.000 (renaming), FLOCK 0.476 (re-scored original); n = 1500 queries; FLOCK: 16 walks, 16 samples									
CoDExSmall s1	T1 inversion	0.493	+4.8%	0.467	0.53	0.485	-0.9%	0.493	-0.02
CoDExSmall s1	control: deleted	0.384	+25.9%	0.312	0.69	0.457	+4.8%	0.266	0.45
CoDExSmall s1	T2 reified	0.382	+26.2%	0.385	0.62	0.408	+15.1%	0.414	0.14
reference EVC: ULTRA 1.000 (renaming), FLOCK 0.484 (re-scored original); n = 1500 queries; FLOCK: 16 walks, 16 samples									
DBpedia100k s0	T1 inversion	0.347	+8.4%	0.470	0.53	0.314	-1.3%	0.668	0.03
DBpedia100k s0	control: deleted	0.225	+40.8%	0.352	0.65	0.258	+16.9%	0.346	0.50
DBpedia100k s0	T2 reified	0.184	+51.5%	0.251	0.75	0.281	+9.5%	0.639	0.07
reference EVC: ULTRA 0.999 (renaming), FLOCK 0.689 (re-scored original); n = 800 queries; FLOCK: 16 walks, 8 samples									
DBpedia100k s1	T1 inversion	0.369	+2.7%	0.455	0.54	0.324	+1.4%	0.636	0.02
DBpedia100k s1	control: deleted	0.227	+40.2%	0.371	0.63	0.268	+18.5%	0.360	0.44
DBpedia100k s1	T2 reified	0.187	+50.7%	0.271	0.73	0.300	+8.9%	0.595	0.08
reference EVC: ULTRA 0.996 (renaming), FLOCK 0.646 (re-scored original); n = 800 queries; FLOCK: 16 walks, 8 samples									
Hetionet s0	T1 inversion	0.266	-12.5%	0.147	0.85	0.091	+2.9%	0.407	-0.02
Hetionet s0	control: deleted	0.279	-18.1%	0.192	0.81	0.131	-39.3%	0.228	0.43
Hetionet s0	T2 reified	0.160	+32.2%	0.205	0.80	0.092	+2.3%	0.409	-0.02
reference EVC: ULTRA 1.000 (renaming), FLOCK 0.400 (re-scored original); n = 800 queries; FLOCK: 16 walks, 8 samples									
WN18RR s0	T1 inversion	0.295	+13.6%	0.296	0.70	0.425	-1.3%	0.779	0.01
WN18RR s0	control: deleted	0.194	+43.3%	0.155	0.84	0.305	+27.2%	0.357	0.55
WN18RR s0	T2 reified	0.221	+35.3%	0.198	0.80	0.391	+6.7%	0.675	0.14
reference EVC: ULTRA 1.000 (renaming), FLOCK 0.789 (re-scored original); n = 800 queries; FLOCK: 16 walks, 8 samples									

Table 12: SchemaPetals: PETALS instances of Kim et al. (2026) with their petal relations reified (context-only; 220 instances, seed 0). "trained" means fine-tuned on half of the original instances and tested on the held-out half in both encodings. Neither model solves PETALS zero-shot, so only the trained rows are informative: trained FLOCK solves 100% of the held-out original instances and 98.2% of their reified versions, trained ULTRA 43.6% and 30.9%.

Setting	Model	PETALS	SchemaPetals	decision flips
zeroshot	ULTRA	22.7%	15.0%	32.3%
zeroshot	FLOCK	51.8%	50.9%	52.7%
trained	ULTRA	43.6%	30.9%	50.9%
trained	FLOCK	100.0%	98.2%	1.8%

E.2 LEARNED REMEDIES

Table 13: Learned objectives versus augmentation on held-out rewrite families (six CoDExSmall and two DBpedia100k fold/seed runs; mean ± standard deviation across runs). "consistency" is augmentation plus the consistency term; "consistency + margin" adds the margin term. The paired per-run differences quoted in §6.1 are computed run by run and differ from differences of these column means only by rounding.

Config	original	T2 reified seen	HELDOUT mixed	HELDOUT roleshare
CoDExSmall, 6 fold/seed runs				
zero-shot (n=6)	0.553 ± 0.210	0.398 ± 0.118	0.441 ± 0.145	0.410 ± 0.117
augmentation (n=6)	0.549 ± 0.205	0.451 ± 0.142	0.462 ± 0.146	0.429 ± 0.123
consistency (n=6)	0.542 ± 0.192	0.434 ± 0.122	0.449 ± 0.129	0.428 ± 0.121
consistency + margin (n=6)	0.546 ± 0.188	0.445 ± 0.133	0.463 ± 0.140	0.432 ± 0.122
DBpedia100k, 2 fold/seed runs				
zero-shot (n=2)	0.511 ± 0.007	0.212 ± 0.019	0.261 ± 0.046	0.235 ± 0.023
augmentation (n=2)	0.501 ± 0.028	0.356 ± 0.022	0.360 ± 0.030	0.346 ± 0.018
consistency (n=2)	0.497 ± 0.019	0.360 ± 0.004	0.363 ± 0.024	0.349 ± 0.008

Table 14: View-specific fine-tuning (single folds, seed 0), MRR. Columns: A, zero-shot on the original view; B, zero-shot on the reified view; C, fine-tuned on the original view and evaluated there; D, fine-tuned on the reified view and evaluated there, with matched steps. $D - B$ is the part of the gap recovered by adaptation; $C - D$ is the residual gap after matched fine-tuning, with its 95% paired bootstrap interval.

run	A	B	C	D	$D - B$	$C - D$ [95% CI]
CoDExSmall f0	0.298	0.263	0.322	0.283	+0.020	+0.039 [+0.027, +0.051]
CoDExSmall f1	0.799	0.527	0.828	0.802	+0.275	+0.026 [+0.014, +0.039]
CoDExSmall f2	0.574	0.453	0.596	0.564	+0.111	+0.032 [+0.016, +0.049]
DBpedia100k f0	0.295	0.137	0.270	0.229	+0.092	+0.041 [+0.022, +0.059]
DBpedia100k f1	0.516	0.225	0.530	0.407	+0.181	+0.123 [+0.103, +0.145]
DBpedia100k f2	0.456	0.194	0.467	0.320	+0.125	+0.148 [+0.127, +0.168]

Three features of Fig. 9 deserve comment. Augmentation transfers to unseen depths but stays well below the original view. One normalizer pass removes one rewrite level: on CoDExSmall, where no genuine node is collapsed, its curve is the zero-shot curve shifted by one depth, whereas on DBpedia100k and WN18RR a single pass also collapses genuine nodes, so its depth-1 MRR stays below the original (0.527 vs 0.531; 0.583 vs 0.647). The iterated support-only normalizer, which did not converge on DBpedia100k, slightly exceeds the original MRR there (0.538 vs 0.531) while changing 22–23% of top-1 answers (EVC 0.77–0.78); the iterated purity-filtered normalizer returns near-original MRR at every depth on all three graphs.

Table 15: Composition: trained on depth 1, tested at depths 2–3, versus one-pass and iterated normalization (fold 1, seed 0; all folds and seeds in Table 16). Entries are MRR / EVC on the fold's queries; depths ≥ 2 are never seen in training; normalizer rows use the zero-shot model. "Iterated" rows repeat the pass until none collapse, with a six-pass cap (Appendix B); * marks runs that reached the cap without converging (support-only on DBpedia100k and WN18RR), unmarked iterated rows converged (support-only on CoDExSmall; every purity-rule row), and "yield rule" rows stopped when a pass collapsed fewer than 5% of the nodes of the previous pass. The "original" column is the fold-1 MRR, which differs from the pooled value of Table 1.

method	original	depth 1 (seen)	depth 2	depth 3
AristoV4				
zero-shot	0.506	0.325 / 0.12	0.250 / 0.07	0.250 / 0.08
normalizer, one pass	–	0.510 / 0.92	0.326 / 0.11	0.252 / 0.07
iterated, 5% yield rule	–	0.510 / 0.92	0.510 / 0.93	0.510 / 0.93
iterated, purity rule	–	0.509 / 0.93	0.509 / 0.94	0.509 / 0.94
iterated, purity rule, support 1	–	0.508 / 1.00	0.508 / 1.00	0.508 / 1.00
CoDExSmall				
zero-shot	0.799	0.527 / 0.32	0.496 / 0.29	0.495 / 0.30
augmentation	0.819	0.634 / 0.47	0.625 / 0.49	0.624 / 0.49
+ consistency	0.783	0.564 / 0.46	0.563 / 0.45	0.563 / 0.45
+ consistency + margin	0.791	0.606 / 0.58	0.601 / 0.52	0.601 / 0.52
normalizer, one pass	–	0.799 / 1.00	0.527 / 0.32	0.496 / 0.29
normalizer, iterated (support only)	–	0.799 / 1.00	0.799 / 1.00	0.799 / 1.00
iterated, 5% yield rule	–	0.799 / 1.00	0.799 / 1.00	0.799 / 1.00
iterated, purity rule	–	0.799 / 1.00	0.799 / 1.00	0.799 / 1.00
DBpedia100k				
zero-shot	0.531	0.224 / 0.23	0.166 / 0.20	–
augmentation	0.527	0.381 / 0.47	0.264 / 0.42	–
+ consistency	0.513	0.372 / 0.49	0.255 / 0.43	–
normalizer, one pass	–	0.527 / 0.94	0.222 / 0.24	–
normalizer, iterated (support only)*	–	0.538 / 0.77	0.538 / 0.78	–
iterated, 5% yield rule	–	0.527 / 0.94	0.528 / 0.94	–
iterated, purity rule	–	0.531 / 1.00	0.531 / 1.00	–
Hetionet				
zero-shot	0.236	0.188 / 0.03	0.188 / 0.04	–
normalizer, one pass	–	0.236 / 1.00	0.188 / 0.03	–
iterated, purity rule	–	0.236 / 1.00	0.236 / 1.00	–
WN18RR				
zero-shot	0.647	0.523 / 0.44	0.482 / 0.38	0.483 / 0.38
normalizer, one pass	–	0.583 / 0.67	0.497 / 0.43	0.470 / 0.37
iterated, 5% yield rule	–	0.685 / 0.45	0.685 / 0.45	0.685 / 0.45
iterated, purity rule	–	0.647 / 1.00	0.647 / 1.00	0.647 / 1.00

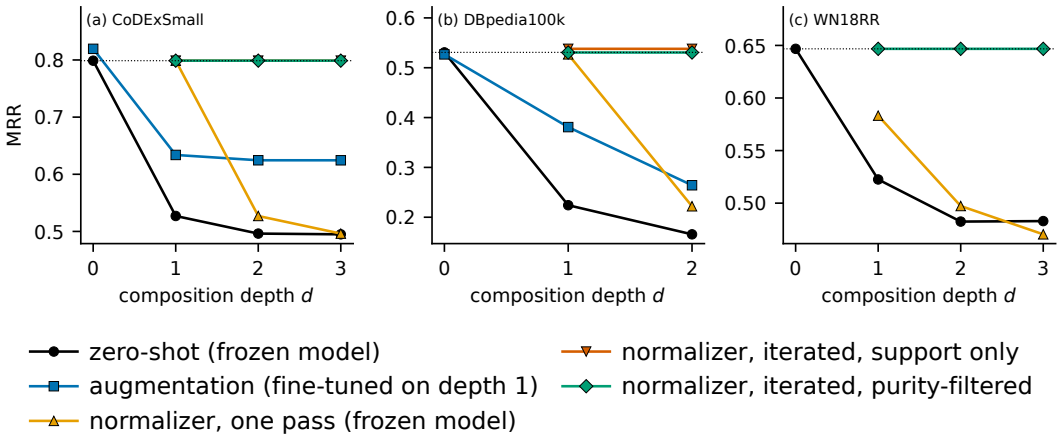


Figure 9: MRR under composed reification (fold 1, seed 0) on (a) CoDExSmall, (b) DBpedia100k, (c) WN18RR: the frozen zero-shot model; augmentation, fine-tuned on depth 1 and tested at unseen depths; and three normalizer settings applied to the frozen model (one pass; iterated, support only; iterated, purity-filtered). The support-only iteration converged on CoDExSmall and stopped at the six-pass cap on DBpedia100k and WN18RR; the purity-filtered iteration converged on every view. WN18RR has no augmentation run. Values and interpretation: Table 15 and the text below.

Table 16: Composition, all runs: MRR / EVC by depth. Iteration rows as in Table 15; * marks support-only iterations that stopped at the six-pass cap (DBpedia100k, WN18RR); the CoDExSmall support-only iteration converged.

run	method	original	depth 1 (seen)	depth 2	depth 3
AristoV4 f1 s0+yield	iterated, yield rule	–	0.510 / 0.92	0.510 / 0.93	0.510 / 0.93
CoDExSmall f1 s0	zeroshot	0.799	0.527 / 0.32	0.496 / 0.29	0.495 / 0.30
CoDExSmall f1 s0	augmentation	0.819	0.634 / 0.47	0.625 / 0.49	0.624 / 0.49
CoDExSmall f1 s0	consistency + margin	0.791	0.606 / 0.58	0.601 / 0.52	0.601 / 0.52
CoDExSmall f1 s0	one pass	–	0.799 / 1.00	0.527 / 0.32	0.496 / 0.29
CoDExSmall f1 s0	iterated (support only)	–	0.799 / 1.00	0.799 / 1.00	0.799 / 1.00
CoDExSmall f1 s0+yield	iterated, yield rule	–	0.799 / 1.00	0.799 / 1.00	0.799 / 1.00
CoDExSmall f1 s0 (pc)	zeroshot	0.799	0.527 / 0.32	0.496 / 0.29	0.495 / 0.30
CoDExSmall f1 s0 (pc)	consistency	0.783	0.564 / 0.46	0.563 / 0.45	0.563 / 0.45
CoDExSmall f1 s0 (pc)	one pass	–	0.799 / 1.00	0.527 / 0.32	0.496 / 0.29
CoDExSmall f1 s0 (pc)	iterated (support only)	–	0.799 / 1.00	0.799 / 1.00	0.799 / 1.00
CoDExSmall f1 s1	zeroshot	0.799	0.527 / 0.32	0.496 / 0.29	0.495 / 0.30
CoDExSmall f1 s1	augmentation	0.811	0.622 / 0.44	0.617 / 0.44	0.617 / 0.44
CoDExSmall f1 s1	consistency + margin	0.741	0.517 / 0.46	0.511 / 0.43	0.511 / 0.43
CoDExSmall f1 s1	one pass	–	0.799 / 1.00	0.527 / 0.32	0.496 / 0.29
CoDExSmall f1 s1	iterated (support only)	–	0.799 / 1.00	0.799 / 1.00	0.799 / 1.00
CoDExSmall f1 s1 (pc)	zeroshot	0.799	0.527 / 0.32	0.496 / 0.29	0.495 / 0.30
CoDExSmall f1 s1 (pc)	consistency	0.812	0.564 / 0.45	0.555 / 0.43	0.555 / 0.43
CoDExSmall f1 s1 (pc)	one pass	–	0.799 / 1.00	0.527 / 0.32	0.496 / 0.29
CoDExSmall f1 s1 (pc)	iterated (support only)	–	0.799 / 1.00	0.799 / 1.00	0.799 / 1.00
CoDExSmall f2 s0	zeroshot	0.574	0.453 / 0.39	0.448 / 0.36	0.448 / 0.36
CoDExSmall f2 s0	augmentation	0.573	0.542 / 0.34	0.520 / 0.36	0.520 / 0.36
CoDExSmall f2 s0	consistency + margin	0.578	0.545 / 0.35	0.530 / 0.45	0.529 / 0.45
CoDExSmall f2 s0	one pass	–	0.575 / 0.99	0.453 / 0.39	0.448 / 0.36
CoDExSmall f2 s0	iterated (support only)	–	0.575 / 0.99	0.575 / 0.99	0.575 / 0.99
DBpedia100k f1 s0 (aug)	zeroshot	0.531	0.224 / 0.23	0.166 / 0.20	–
DBpedia100k f1 s0 (aug)	augmentation	0.527	0.381 / 0.47	0.264 / 0.42	–
DBpedia100k f1 s0 (aug)	one pass	–	0.527 / 0.94	0.222 / 0.24	–
DBpedia100k f1 s0 (aug)	iterated (support only)*	–	0.538 / 0.77	0.538 / 0.78	–
DBpedia100k f1 s0+yield	iterated, yield rule	–	0.527 / 0.94	0.528 / 0.94	–
DBpedia100k f1 s0 (pc)	zeroshot	0.531	0.224 / 0.23	0.166 / 0.20	–
DBpedia100k f1 s0 (pc)	consistency	0.513	0.372 / 0.49	0.255 / 0.43	–
DBpedia100k f1 s0 (pc)	one pass	–	0.527 / 0.94	0.222 / 0.24	–
DBpedia100k f1 s0 (pc)	iterated (support only)*	–	0.538 / 0.78	0.538 / 0.77	–
DBpedia100k f2 s0 (aug)	zeroshot	0.448	0.195 / 0.27	0.154 / 0.25	–
DBpedia100k f2 s0 (aug)	augmentation	0.436	0.314 / 0.47	0.212 / 0.43	–
DBpedia100k f2 s0 (aug)	one pass	–	0.443 / 0.89	0.191 / 0.27	–
DBpedia100k f2 s0 (aug)	iterated (support only)*	–	0.445 / 0.77	0.444 / 0.77	–
WN18RR f1 s0+yield	iterated, yield rule	–	0.685 / 0.45	0.685 / 0.45	0.685 / 0.45

E.3 THE NORMALIZER

Table 17: CoDExSmall (five folds pooled, $n = 3{,}656$): blind structural normalizer with one motif (seed-1 run) versus two motifs (seed-0 run). The seed changes the fresh-node identifiers, to which ULTRA is equivariant, so the uniform-reification rows coincide; it also changes the mixed schema's assignment of primitives to relation slices (Appendix C), so the MIXED rows are different rewrite realizations (unnormalized MRR 0.371 vs 0.393) and the two MIXED blocks are an unpaired comparison in which the motif library and the view both differ. "canon@50" withholds a random half of the accepted relation-pair keys (Appendix B), so every instance of a withheld key stays reified. Identical to the oracle-boundary variant on this graph.

View	reification motif only		+ path motif	
	MRR	EVC	MRR	EVC
original	0.472	–	0.472	–
T2 reified	0.355	0.389	0.355	0.389
T2 reified + canon	0.471	0.989	0.471	0.989
T2 reified + canon@50	0.410	0.519	0.419	0.559
MIXED	0.371	0.438	0.393	0.470
MIXED + canon	0.449	0.688	0.471	0.995
MIXED + canon@50	0.413	0.523	0.442	0.620

Table 18: DBpedia100k ($4{,}000$ paired queries per seed): blind versus oracle-boundary normalizer, two seeds, with and without the purity test; "canon@50" withholds half of the accepted relation-pair keys. With the purity test, the blind normalizer matches the oracle within 0.0001 MRR on both views and both seeds. A residual 0.7–1.1% of top-1 answers differ; the 27 single-edge mediators that the support threshold keeps remain in the normalized graph and are the proposed explanation. Adding the consistent conditioning of Prop. 1 leaves EVC at 0.993 on the mixed view, which rules out the conditioning asymmetry as the source but does not isolate the retained mediators as the cause.

View	seed 0		seed 0 + purity		seed 1		seed 1 + purity	
	MRR	EVC	MRR	EVC	MRR	EVC	MRR	EVC
original	0.398	–	0.398	–	0.388	–	0.388	–
T2 reified	0.178	0.250	0.178	0.250	0.176	0.255	0.176	0.255
T2 reified + canon	0.389	0.854	0.398	0.993	0.378	0.836	0.388	0.989
T2 reified + canon@50	0.300	0.433	0.266	0.463	0.312	0.478	0.269	0.467
T2 reified + canon (oracle boundary)	0.398	0.993	0.398	0.992	0.388	0.990	0.388	0.989
MIXED	0.216	0.316	0.216	0.316	0.212	0.327	0.212	0.327
MIXED + canon	0.390	0.825	0.398	0.993	0.377	0.813	0.388	0.990
MIXED + canon@50	0.356	0.531	0.313	0.501	0.348	0.535	0.349	0.547
MIXED + canon (oracle boundary)	0.398	0.992	0.398	0.993	0.388	0.991	0.388	0.990

Table 19: The purity-filtered normalizer under three backbones on CoDExSmall (same folds and queries). On CoDExSmall, normalization recovers MRR to within 0.001 of the original for all three tested backbones, whereas the partial-recognition loss differs by backbone. MOTIF's residual top-1 disagreement at support 2 coincides with three leftover single-edge mediators; at support 1 its MRR and EVC are 0.474 / 1.000.

View	ULTRA		MOTIF		TRIX	
	MRR	EVC	MRR	EVC	MRR	EVC
original	0.472	–	0.474	–	0.473	–
T2 reified	0.355	0.389	0.411	0.284	0.436	0.450
T2 reified + canon	0.471	0.989	0.473	0.938	0.473	0.999
T2 reified + canon@50	0.419	0.559	0.400	0.393	0.453	0.627
MIXED	0.393	0.470	0.428	0.388	0.442	0.521
MIXED + canon	0.471	0.995	0.473	0.967	0.473	1.000
MIXED + canon@50	0.442	0.620	0.415	0.490	0.460	0.699

Table 20: Blind normalization of untouched graphs (ULTRA, $2{,}000$ queries per graph), with a random-collapse control that collapses the same number of degree-2 nodes chosen at random, and with the purity test.

Dataset	original	+ blind canon	δ [95% CI]	random collapse δ	edges moved	+ purity rule	
						δ	EVC
CoDExSmall	0.472	0.472	+0.000 [+0.000, +0.000]	–	0.0%	+0.0000	1.000
ConceptNet100k	0.061	0.160	+0.099 [+0.088, +0.110]	+0.125	3.8%	+0.0000	1.000
WN18RR	0.371	0.346	-0.025 [-0.038, -0.012]	+0.029	21.5%	+0.0000	0.999
DBpedia100k	0.398	0.390	-0.008 [-0.014, -0.002]	-0.019	1.6%	+0.0002	0.999
AristoV4	0.188	0.190	+0.002 [-0.000, +0.004]	+0.002	0.7%	+0.0000	0.999

Table 21: Normalizer false-positive counts on untouched graphs, by motif library and with the relation-purity test (nodes it would collapse; support 2; CPU counts, no model).

Dataset	$\lvert V\rvert$	reification motif		+ path motif		+ purity
		nodes	% edges	nodes	% edges	nodes
CoDExSmall	2,034	0	0.00	0	0.00	0
ConceptNet100k	78,334	902	1.80	1,942	3.88	0
DBpedia100k	99,604	3,288	1.10	4,714	1.58	0
AristoV4	44,949	401	0.33	853	0.70	0
Hetionet	45,158	3	0.00	16	0.00	0
YAGO310	123,182	1,383	0.26	2,656	0.49	0
WN18RR	40,943	1,641	3.78	9,651	22.23	0
FB15k237	14,541	36	0.03	88	0.06	0

Table 22: Additive (mediator kept, collapsed edge added) versus collapsing normalization (ULTRA, five folds pooled, seed 0; reification motif only). The additive variant changes the untouched DBpedia100k graph less, by 0.005 MRR, and is worse on every rewritten view.

Dataset	view	raw	+ destructive canon	+ additive shortcut	shortcut − canon
CoDExSmall	UNTOUCHED	0.472	0.472	0.472	+0.000 [+0.000, +0.000]
CoDExSmall	T2 uniform	0.355	0.471	0.395	-0.076 [-0.084, -0.068]
CoDExSmall	MIXED	0.393	0.448	0.419	-0.029 [-0.034, -0.024]
CoDExSmall	DEPTH2	0.342	0.471	0.240	-0.231 [-0.243, -0.220]
DBpedia100k	UNTOUCHED	0.398	0.390	0.395	+0.005 [+0.002, +0.007]
DBpedia100k	T2 uniform	0.178	0.396	0.311	-0.085 [-0.093, -0.076]
DBpedia100k	MIXED	0.216	0.350	0.313	-0.037 [-0.044, -0.031]
DBpedia100k	DEPTH2	0.148	0.396	0.130	-0.266 [-0.279, -0.254]

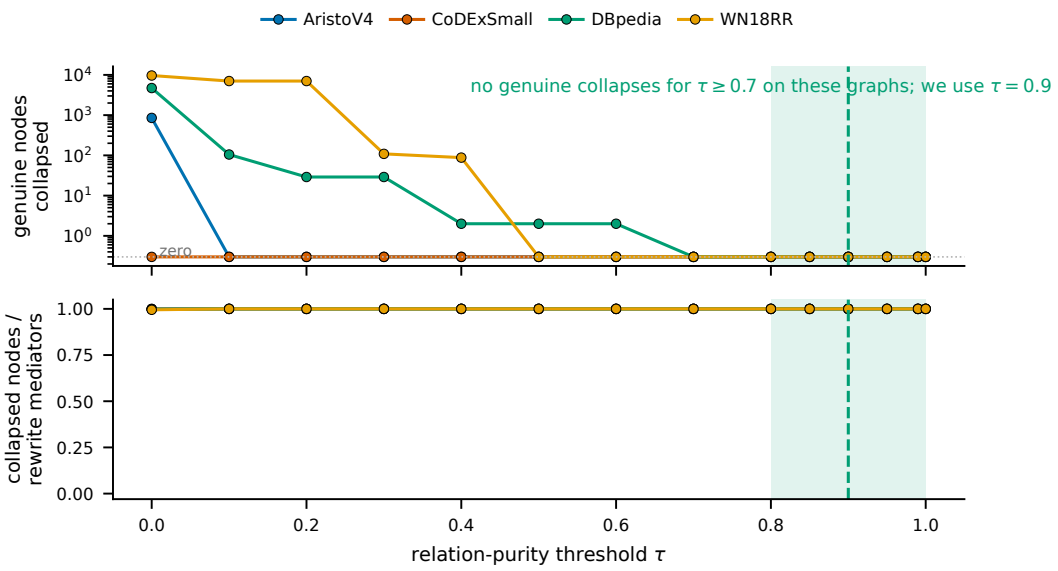


Figure 10: Sensitivity to the purity threshold (CPU counts, no model, support $s = 2$). Top: genuine nodes collapsed on the untouched graph. Bottom: nodes collapsed on the rewritten view divided by the number of mediators the rewrite created; this ratio rounds to 1.00 at two decimals, so the single-edge mediators left standing elsewhere (e.g., 27 of 478,429 on DBpedia100k) do not register at this precision. On the four graphs shown, no genuine node is collapsed for $\tau \geq 0.7$ while the ratio is unchanged, so $\tau = 0.9$ was chosen inside that range and not tuned for accuracy.

Table 23: CPU wall-clock (seconds) of view generation and of one normalizer pass ("canon", "+ purity" and "shortcut" are single-pass timings), and node inflation ($\times|V|$) under context reification. Iterated runs (up to six passes) were not timed separately.

Dataset	$\lvert V\rvert$	$\lvert E\rvert$	reify (s)	$\times\lvert V\rvert$	depth-2 (s)	canon (s)	+ purity (s)	shortcut (s)
CoDExSmall	2,034	32,888	0.2	12.2	1.3	0.9	0.3	0.8
ConceptNet100k	78,334	100,000	1.6	2.1	5.6	0.9	2.0	0.5
WN18RR	40,943	86,835	0.4	2.3	6.5	1.5	1.1	0.8
AristoV4	44,949	242,567	1.9	5.2	16.3	1.9	2.8	1.1
DBpedia100k	99,604	597,572	7.5	5.8	40.8	4.5	7.4	2.8
FB15k237	14,541	272,115	2.1	15.4	17.0	2.5	3.9	1.4
YAGO310	123,182	1,079,040	10.0	6.7	66.3	6.9	12.6	7.5
Hetionet	45,158	2,025,177	22.8	34.7	139.0	17.1	19.9	12.5

F Scope and Controls

This appendix summarizes, in one place, what each design choice controls for and where the corresponding evidence is.

- *Query difficulty.* The context-only protocol never rewrites the query relation; the query, candidate set, number of missing facts, and filtered set are identical across views (§3). Global inversion is the one exception and is validated with the flipped query.
- *Graph size.* Fresh nodes are never candidates, and the matched-noise condition adds the same number of nodes and edges with the same mediator degree profile (§5.1); it does not control every topological difference.
- *Synthetic rewrites.* The Freebase-style expansion of FB15k-237 recreates a Freebase-style mediator encoding from real collapsed relations and costs 16.8–17.8% against 19.0–20.1% for the synthetic reification of the same relations (Table 2).

- *Breadth.* Seven graphs, three message-passing backbones, one walk-based backbone at a reduced budget, and complex logical queries (§5); learned remedies are evaluated on ULTRA only.
- *Normalization.* Motif detection uses graph structure alone, without a rewrite manifest or labels identifying introduced mediator nodes; separately, the evaluation procedure retains the original candidate entities and maps query relations through the returned relation mapping (Appendix B). The normalizer is compared against an oracle-boundary variant that does receive the labels identifying introduced mediators (Table 18); its guarantee is conditional (Prop. 2) and its measured failures are partial recognition, single-edge mediators, and unknown motifs (§6).
- *Augmentation.* Augmentation is the baseline for the learned objectives, not a competitor omitted from them (Table 13).
- *Equivariance.* Link invariance is defined within $\mathcal{K}_{n,m}$; renaming stays inside it and is free (Prop. A.1), reification leaves it (Prop. A.2).
- *Equivalence.* Answer preservation is defined relative to a declared query family and checked by an executable validator for every evaluation query on the observed graph (§3).