KNOWLEDGE PROPERTY LEARNING AND EVALUATION IN LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Knowledge editing corrects isolated facts in large language models, yet facts do not live in isolation: symmetric, inverse and transitive relations constrain many facts at once, and an edited model that violates them holds inconsistent knowledge. We formalise this gap as *property learning*: encoding a relational property through editing so that the model applies it to facts it was never shown. We introduce **PropEdit-Bench**, spanning geospatial and social domains, which unlike existing benchmarks assesses property abstraction directly: properties are injected on one set of entities and tested on a disjoint one. Property-aware metrics complement completion with binary judgements in both polarities, making the evaluation robust to one-to-many answers. Across LLM families and scales, a small number of property-instantiating edits, applied before the new fact, raise compliance over standard editing in every model, property and domain for individually edited facts, by up to 55% relative. Completion and binary evaluations frequently disagree, exposing relations that models recognise but do not produce. Factual correction and relational structure are thus distinct capabilities, calling for editing methods that explicitly target the structure underlying edited knowledge.

1 INTRODUCTION

Large language models encode vast amounts of world knowledge during pre-training, yet this knowledge is static, becomes outdated and contains errors that are expensive to correct through retraining. Knowledge editing has emerged as a targeted alternative that updates specific facts directly in the model weights (De Cao et al., 2021; Mitchell et al., 2022a; Meng et al., 2022). Current methods frame each update as an isolated triplet (s, r, o) and optimise single-fact or batch injection, judged by whether the edited fact, its paraphrases and unrelated facts behave as intended (Meng et al., 2023; Fang et al., 2025; Yao et al., 2023). This framing ignores a fundamental dimension of relational knowledge: the structural properties that relations obey.

Relations are not arbitrary mappings between entities. For example, in geographical knowledge, the symmetric relation `borders` requires $\texttt{borders}(A, B) \Rightarrow \texttt{borders}(B, A)$, a pair such as (`north`, `south`) is reversible, and `within` is transitive. Nothing in current editing pipelines ensures that a model edited to store $\texttt{north}(A, B)$ will also infer $\texttt{south}(B, A)$, and the resulting knowledge is internally inconsistent. Prior work documents this failure: models trained on "A is B" do not infer "B is A" (Berglund et al., 2024), and edits rarely propagate to reverse, logically implied or multi-hop facts (Ma et al., 2023; Zhong et al., 2023; Cohen et al., 2024; Li et al., 2024). These studies measure whether a single edit propagates, on the entities that were edited, and their benchmarks, drawn from encyclopaedic sources, conflate relational structure with domain-specific content. They cannot establish whether a property can be *learned* through editing and applied to entities never seen during supervision.

We argue that property learning, the explicit encoding of relational properties through editing, should be treated as a first-class objective in knowledge editing, distinct from and complementary to fact-level injection. The challenge is structural: a locate-then-edit method writes each fact under its own key, so a fact and its counterpart are two unrelated associations, and a property can only emerge if an update fitted on some keys generalises to keys it has never seen. To tackle this challenge, we introduce **PropEdit-Bench**, spanning geo-spatial and social relations that share the same relational structure under different surface content. The benchmark is designed to enable a

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.

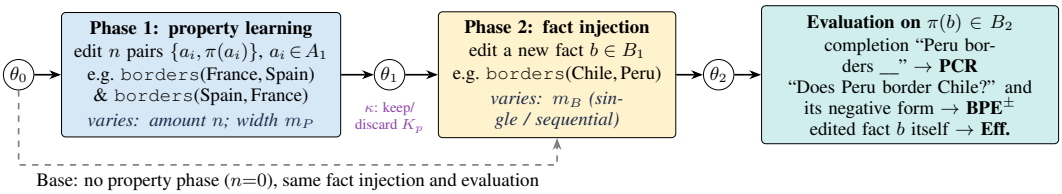


Figure 1: Property learning protocol. Entities are split into disjoint sets A and B, and π maps a fact to its counterpart under the property (here symmetry). The model is never shown $\pi(b)$ or any pair instantiating the property on B entities, so compliance on $\pi(b)$ beyond Base can only come from the property phase. All conditions share the same fact injection and evaluation; they differ in what precedes it and in whether the property fact edits are preserved afterwards (κ). Joint (not drawn) merges both phases into one update.

property learning protocol (Figure 1) in which property-instantiating pairs are first edited on one set of entities, after which a new fact is edited on a disjoint set of entities and the *edited model* is evaluated on whether it produces the relational counterpart of that fact. Comparing this setting with the same edit performed without property supervision isolates the contribution of the relational property to the model's behavior. Property-aware metrics score each edited fact jointly with its counterpart. Because relational queries are one-to-many, a completion naming another valid entity would count as a failure, so we also ask each counterpart as a yes/no question, in a positive and a negative form to guard against a model that affirms everything. Comparing the two formats then reveals whether a model that recognises a relation also produces it.

2 Problem Setting and Research Questions

Relational properties. A fact is a triplet $a = (s, r, o)$ whose subject s and object o are, in our context *entities*: geographic locations in the geo-spatial domain and persons in the social domain. A relational property derives a second fact from the first, written as a map π. *Symmetry* gives $\pi(s, r, o) = (o, r, s)$; *reversibility* gives $\pi(s, r, o) = (o, r^{-1}, s)$ for inverse pairs; *transitivity* derives (s, r, o) from (s, r, m) and (m, r, o). We call $\pi(a)$ the *counterpart* of a; a model is *compliant* on a if, having learned a, it also produces $\pi(a)$. A symmetric counterpart reuses the relation, whereas a reverse counterpart switches to another relation, the setting of the reversal curse (Berglund et al., 2024; Ma et al., 2023).

Property learning and evaluation. We study property learning as a two-step process. In *property learning*, the model is taught a property through fact injection: pairs $\{a_i, \pi(a_i)\}$ that instantiate it are written for one set of entities. A new fact b about other entities is then injected, and *property evaluation* measures whether the model also produces its counterpart $\pi(b)$, which it has never seen. Applied without property learning (Base), the same evaluation gives the compliance achieved by standard editing alone, so the difference between the two quantifies the effect of property learning.

Research questions. Our study is organised around three questions, each answered at the end of the corresponding subsection of §6.

- **RQ1 (Property abstraction).** Does editing pairs that instantiate a property on one set of entities raise compliance on new facts about a disjoint set, beyond what editing the new fact alone achieves? In other words, is the property abstracted, or are only the edited pairs memorised? (§6.1)
- **RQ2 (Supervision scale and injection mode).** How does compliance depend on the number of supervised pairs, on whether they are written in one update or in chunks, on whether they precede or accompany the new fact, and on whether the property edits are protected from later updates? (§6.2)
- **RQ3 (Consistency across evaluation formats).** Do completion-based and binary judgements of the same property agree, before and after property learning? (§6.3)

3 Related Work

Knowledge editing. Parameter-preserving editors store edits outside the base weights: in a retrieval memory (SERAC; Mitchell et al., 2022b), in the prompt (IKE; Zheng et al., 2023), in a discrete key–value codebook (GRACE; Hartvigsen et al., 2023) or in a routed side memory (WISE; Wang et al., 2024). Parameter-modifying editors either learn the update with a hypernetwork (De Cao et al., 2021; Mitchell et al., 2022a) or locate and write it. ROME (Meng et al., 2022) uses causal tracing to find mid-layer MLPs that recall a fact and writes one fact as a rank-one update; MEMIT (Meng et al., 2023) spreads a least-squares update over several layers to edit thousands of facts at once, and EMMET (Gupta et al., 2024b) shows that both optimise the same preservation–memorisation objective. Sequential editing degrades the model gradually and then abruptly (Gupta et al., 2024a); PRUNE (Ma et al., 2025) bounds the perturbation and AlphaEdit (Fang et al., 2025), which we build on, projects it onto the null space of preserved knowledge. Localisation does not always predict where an edit works best (Hase et al., 2023), and editing can distort the structure of what the model knows (Li et al., 2024; Nishi et al., 2025).

Relational properties and editing. Models trained on "A is B" do not learn "B is A" (Berglund et al., 2024; Allen-Zhu & Li, 2025); reverse training reduces this failure (Golovneva et al., 2024), and models that fail to generate the reverse often recognise it among options (Lin et al., 2024). For editing, BAKE (Ma et al., 2023) shows that ROME, MEMIT and fine-tuning do not propagate edits to the reverse direction, and its BIRD method adds a bidirectional objective to each single edit; logically related edits can also conflict (Li et al., 2024). *Symmetry.* Niu et al. (2024) use symmetry as a criterion for evaluating edits under the knowledge-neuron view, and Yuan & Vlachos (2025) find that pretrained encoders are near chance on symmetric and antisymmetric relations, then retrain the encoder with a RotatE-style (Sun et al., 2019) contrastive objective. *Composition and transitivity.* MQuAKE (Zhong et al., 2023) shows that edits rarely carry over to multi-hop questions, and RippleEdits (Cohen et al., 2024) tests logical generalisation (including symmetric relations such as `sibling`) and compositionality, finding that existing editors rarely make these dependent changes. Relations are often decoded by an approximately linear map (Hernandez et al., 2024), and in small models trained on a synthetic family graph, a bilinear relational geometry lets edits reach inverse facts (Kim et al., 2025).

Evaluation benchmarks and metrics. Standard editing benchmarks, ZsRE (Levy et al., 2017) and CounterFact (Meng et al., 2022), evaluate the efficacy, paraphrase generalisation and locality of the edited fact itself. Later benchmarks probe its consequences: portability (Yao et al., 2023) adds reversed-relation and one-hop queries, while BAKE, MQuAKE, RippleEdits and ConflictEdit (Li et al., 2024) target reverse, multi-hop, ripple and conflicting edits (Table 1). Three limitations prevent these benchmarks from measuring whether a relational property is learned. First, consequences are evaluated on the edited entities themselves, so property learning cannot be distinguished from the memorisation of edited pairs. Second, being drawn from encyclopaedic sources with mostly counterfactual targets, they conflate relational structure with domain content and with conflicts against pre-training, and each covers only one or two properties; none evaluates an injected property on held-out entities. Third, their metrics score a single expected completion, although relational queries are inherently one-to-many, and scores depend on the answer format: consistency across paraphrases is not guaranteed (Elazar et al., 2021), generation and recognition of a fact can disagree (Lin et al., 2024), and locality is sensitive to how it is measured (Liu et al., 2026).

4 Main Contributions

We study knowledge editing through a *property-based* lens, shifting the unit of analysis from isolated facts to the relational properties that govern entire classes of facts. Our main contributions are as follows.

1. **A systematic property learning framework.** We formalise property learning as a first-class editing objective, distinct from and complementary to fact-level injection, with a controlled two-phase protocol whose disjoint supervision and evaluation entities ensure that any gain reflects property abstraction rather than memorisation. Spanning single, batched and sequential injec-

Table 1: Comparison of related work on relational-property coverage and injection-mode granularity. ✓ = addressed; ✗ = not addressed; ~ = partial or indirect; — = not applicable. **Eval**: property used as an evaluation criterion. **Inj**: the work explicitly attempts to inject the property into the model. Injection modes: **Single** = single-fact injection; **S-Batch** = simultaneous-batch injection; **Seq-S** = sequential single-fact injections; n = variable batch size.

Work		Symmetry		Reverse		Transitivity		Injection mode			
		Eval	Inj	Eval	Inj	Eval	Inj	Single	S-Batch	Seq-S	n
Berglund et al. (2024)	*Reversal curse*	✗	✗	✓	✗	✗	✗	—	—	—	—
Yuan & Vlachos (2025)	*Sym.-aware training*	✓	~	✗	✗	✗	✗	—	—	—	—
Niu et al. (2024)	*KN thesis*	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗
Ma et al. (2023)	*BAKE / BIRD*	✗	✗	✓	✓	✗	✗	✓	✗	✗	✗
Li et al. (2024)	*ConflictEdit*	✗	✗	✓	✗	~	✗	✓	✗	✗	✗
Cohen et al. (2024)	*RippleEdits*	~	✗	✓	✗	~	✗	✓	✗	✗	✗
Zhong et al. (2023)	*MQuAKE*	✗	✗	✗	✗	✓	✗	✓	✗	✗	✗
Ours	*PropEdit-Bench*	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

tion, it provides a unified framework to analyse property learning, interference and forgetting (§5.2).

2. **PropEdit-Bench.** A property-guided benchmark that addresses the limitations of existing editing benchmarks (§3). It spans geospatial relations of topology, distance and direction and kinship relations from Wikidata5m (Wang et al., 2020), two domains that share relational structure under different surface semantics. It covers symmetry, reversibility and transitivity, with collision-free counterparts, paraphrases and binary formulations in both polarities (§5.3).
3. **Property-aware evaluation metrics.** A robust evaluation centred on a property-level measure, the property-compliance rate, which scores each edited fact jointly with its counterpart, complemented by a binary property evaluation in both polarities and by standard efficacy, generalisation and specificity metrics. The evaluation addresses the one-to-many challenge inherent in relational data, guards against affirmation bias, and probes whether injected properties integrate with and generalise beyond pre-existing model knowledge (§5.4).
4. **Systematic ablation study.** Controlled ablations over model family and scale, domain, supervision scale and injection mode characterise when property learning improves compliance on unseen entities, how much supervision it requires, and how its measurement depends on the answer format (§6).

5 Methodology

5.1 Preliminaries: Locate-then-Edit

Facts as key–value associations. Following Geva et al. (2021) and Meng et al. (2022), the output projection $W \in \mathbb{R}^{d_1 \times d_0}$ of an MLP block is read as a linear associative memory. For a fact (s, r, o), the input to W at the last subject token of the prompt formed by (s, r) is a *key* $k \in \mathbb{R}^{d_0}$ that encodes the subject in the context of the relation. The editor optimises a *value* $v \in \mathbb{R}^{d_1}$, the output of W at that position that makes the model produce o, and then changes W so that $Wk \approx v$. For u facts edited together, keys and values are stacked as columns of $K_1 \in \mathbb{R}^{d_0 \times u}$ and $V_1 \in \mathbb{R}^{d_1 \times u}$. Knowledge to preserve is summarised by keys K_0 collected on generic text, whose values are $V_0 = WK_0$, so preserving it amounts to keeping ΔK_0 small. MEMIT (Meng et al., 2023) solves

$$\Delta = \arg\min_{\tilde{\Delta}} \|(W + \tilde{\Delta})K_1 - V_1\|^2 + \|\tilde{\Delta}K_0\|^2 = R\,K_1^\top \left(K_0 K_0^\top + K_1 K_1^\top\right)^{-1}, \tag{1}$$

with residual $R = V_1 - WK_1$.

AlphaEdit. Eq. 1 trades the two terms off, so repeated edits gradually overwrite preserved knowledge. AlphaEdit (Fang et al., 2025) projects the update with P, the projector onto the null space of $K_0 K_0^\top$, so that preserved outputs are unchanged by construction. For sequential editing, it also retains the keys K_p of earlier edits:

$$\Delta_{\text{AlphaEdit}} = R\,K_1^\top P \left(K_p K_p^\top P + K_1 K_1^\top P + I\right)^{-1}. \tag{2}$$

Algorithm 1 Two-phase injection: property learning, then fact injection.

Require: θ_0, K_0, pairs (A_1, A_2), B_1, n, m_P, m_B, κ
1: $\theta \leftarrow \theta_0$; $K_p \leftarrow \emptyset$; $\mathcal{P} \leftarrow \textsc{Stratify}(A_1, n)$
2: **for all** chunk C of $\mathcal{P}$ of size m_P **do** ▷ property learning
3: $S \leftarrow \bigcup_{i \in C} \{a_i, \pi(a_i)\}$ ▷ pairs intact
4: $\theta, K_p \leftarrow \textsc{Edit}(\theta, S; K_0, K_p)$
5: **end for**
6: $\theta_1 \leftarrow \theta$; $K_p^{(1)} \leftarrow (\kappa{=}\textsf{keep})\ ?\ K_p : \emptyset$
7: **for all** $b \in B_1$ **do** ▷ fact injection
8: **if** $m_B = 1$ **then** $\theta \leftarrow \theta_1$; $K_p \leftarrow K_p^{(1)}$
9: **end if**
10: $\theta, K_p \leftarrow \textsc{Edit}(\theta, \{b\}; K_0, K_p)$
11: evaluate b and its counterpart $\pi(b) \in B_2$
12: **end for**

Algorithm 2 Joint injection: property pairs and the new fact in one update.

Require: θ_0, K_0, pairs (A_1, A_2), B_1, n
1: $\mathcal{P} \leftarrow \textsc{Stratify}(A_1, n)$
2: **for all** $b \in B_1$ **do**
3: $S \leftarrow \bigcup_{i \in \mathcal{P}} \{a_i, \pi(a_i)\} \cup \{b\}$
4: $\theta \leftarrow \textsc{Edit}(\theta_0, S; K_0, \emptyset)$
5: evaluate b and $\pi(b)$
6: **end for**

Base is Alg. 1 with $n{=}0$. Fresh θ_0 per trial; no phase boundary, so no κ.

The K_p term preserves the outputs of earlier edits in the same way as the least-squares objective preserves knowledge, while K_0 is enforced by projection. AlphaEdit improves most locate-then-edit methods, and we use it as the editor throughout.

5.2 Property learning

Design principle: a property must generalise to unwritten keys. An editor writes associations, not rules. Under Eq. 2, a fact and its counterpart are read from different keys, since their subject and relation differ, so writing a pair $\{a, \pi(a)\}$ stores two independent associations. The key of the counterpart $\pi(b)$ of a new fact is never written: the edited layer reaches it only through Δk, a mix of the residuals of the edited facts weighted by key similarity. A relational property therefore cannot be stored by an editor; it can only emerge when an update fitted on some keys generalises to keys it has never seen. **This principle sets three requirements for our protocol:** (i) evaluation must target keys that were never written, (ii) property learning should precede the new fact so that it can shape the weights first, and (iii) a rule internalised in the model must be distinguishable from the examples that taught it.

Protocol. Algorithm 1 meets these requirements in turn. Entities are partitioned into disjoint sets A and B, giving base facts and counterparts A_1, $A_2{=}\pi(A_1)$ and B_1, $B_2{=}\pi(B_1)$ (App. E.1), and evaluation queries only B_2 (l. 11), whose keys are never written. Property learning comes first (ll. 2–4): n pairs from (A_1, A_2), sampled equally per relation (l. 1), are written in batches of m_P, each pair kept within one update since neither half instantiates the property alone (l. 3). Fact injection follows (ll. 7–11): each new fact $b \in B_1$ is written either on a fresh copy of θ_1 ($m_B{=}1$, l. 8) or after the previous ones ($m_B{=}$seq). At the phase boundary (l. 6), κ decides whether the keys K_p of the learned pairs constrain the fact updates (l. 10). With keep, writing b must preserve the teaching examples; with discard, they are released, and compliance must rest on the regularity written into W. A model that has internalised the rule no longer needs its examples. Both settings start from the same θ_1, so their comparison is controlled; since discard also relaxes the constraints on the fact update, we read it together with supervision scale and sequential editing (§6.2, App. C). Base runs the same algorithm with $n{=}0$, and each condition is compared with the Base run sharing its m_B, so that compared conditions differ by a single factor; n is swept from 10 to $|A_1|$, restarting from θ_0 at each value.

5.3 PropEdit-Bench

PropEdit-Bench contains two domains with the same relational structure and different surface content. The **Geo** domain relates geographic locations through eight spatial relations: `close`, `far` and `borders` (symmetric), `north`/`south` and `east`/`west` (reversible). It is drawn from a base of 255,530 location pairs that we constructed for this work using YAGO2GEO (Karalis et al., 2019). The **Wiki** domain relates persons through kinship relations from Wikidata5m (Wang et al., 2020):

`sibling` and `spouse` (symmetric) and `father`/`child` (reversible), subsampled to a size comparable to Geo. Each subject–relation pair has exactly one object, every fact has a positive and a negative binary question, objects are diversified, and each fact comes with paraphrases and specificity probes. Construction, splits, statistics and a comparison with existing datasets are in App. D–E.

5.4 Evaluation metrics

Efficacy (Eff.), paraphrase generalisation and specificity follow the editing literature (Meng et al., 2022; Yao et al., 2023) and are defined in App. F; we introduce two property-level metrics.

Property-compliance rate (PCR). A counterpart can only be expected once its fact is learned, so PCR scores each edited fact jointly with its counterpart over the N test pairs $(b_t, \pi(b_t))$:

$$\mathrm{PCR} = \frac{1}{N} \sum_{t=1}^{N} s(b_t)\, s\big(\pi(b_t)\big), \tag{3}$$

where $s \in [0, 1]$ is a logit-based score of the gold answer. PCR admits thresholded and conditioned variants that capture complementary facets of property compliance (App. F).

Binary property evaluation (BPE). Relational queries are one-to-many (a location borders several others), so a completion naming another valid entity is wrongly scored as a failure. BPE instead probes the specific counterpart with a yes/no question, in a positive form q_t^+ (answer *yes*) and a negative form q_t^- (opposite relation for Geo, negated relation for Wikidata; answer *no*), so that a model affirming everything is not rewarded:

$$\mathrm{BPE}^{\pm} = \frac{1}{|\mathcal{L}|} \sum_{t \in \mathcal{L}} \mathbb{1}\big[\ell_{yes}(q_t^{\pm}) > \ell_{no}(q_t^{\pm})\big], \tag{4}$$

where ℓ_{yes} and ℓ_{no} are the logits of the answer tokens and $\mathcal{L}$ is the set of test pairs whose edited fact is learned (App. F). A compliant model combines a high BPE^+ with a low BPE^-; **Port.** is defined identically on questions about the edited fact b_t.

6 Experiments and Results

Setup. We compare batch property learning, completed before any new fact in one update (Batch) or in chunks (Chunk), and joint injection (Joint) with Base, which measures the compliance obtained through fact editing alone (§5.2). We edit LLMs of two families and several scales, Llama3-8B, Llama3-70B (Grattafiori et al., 2024), Qwen3-32B and Qwen3-8B (Yang et al., 2025), with AlphaEdit; Qwen3-8B, the Wikidata tables, efficacy and the remaining conditions are in App. B, and the sequential fact-injection regime (seq B_1 rows) is analysed in App. C. Table 2 reports, for each cell, the scale point of the grid (App. Table 3) with the highest PCR, and Figures 2 and 3 show the full sweeps. Since PCR is computed on learned facts, its estimate is more precise when efficacy is high, which holds in most settings, where Eff. exceeds 60% (Table 2); the few exceptions are noted where relevant. Transitivity experiments are in App. G.

6.1 RQ1: Property abstraction

Property supervision transfers to unseen entities. When new facts are edited individually, property supervision improves PCR over the matched Base for all three models, both properties, both corpora and both values of κ, and in most cases for Qwen3-8B (App. B). On symmetry, Batch raises PCR from 21.18 to 32.93 on Llama3-8B (+55%)[1], from 34.78 to 37.01 on Llama3-70B (+6%) and from 25.78 to 28.82 on Qwen3-32B (+12%); reversibility shows the same pattern, with gains of +41%, +10% and +5%. Measured on entities absent from every supervised pair, these gains show that the edits change how the model treats the relation itself. Property learning also helps acquire the new fact: efficacy rises over Base in different Geo settings (e.g. 72.14 → 96.31 for Llama3-8B

[1] Relative improvement (new − old)/old.

Prop.	Mode	$\mathcal{K}_p$	Llama3-8B PCR ↑	Llama3-8B BPE (+/−) ↑	Llama3-8B Port. ↑	Llama3-8B Eff. ↑	Llama3-70B PCR ↑	Llama3-70B BPE (+/−) ↑	Llama3-70B Port. ↑	Llama3-70B Eff. ↑	Qwen3-32B PCR ↑	Qwen3-32B BPE (+/−) ↑	Qwen3-32B Port. ↑	Qwen3-32B Eff. ↑
Symmetry	baseline		21.18	98.61/1.39	99.31/2.78	72.14	34.78	45.83/53.47	60.42/70.83	94.61	25.78	54.17/19.44	75.69/32.64	85.96
	single_prop		17.99	92.08/3.52	96.19/9.68	**50.23**	**38.57**	40.32/64.52	**58.06/79.03**	89.91	24.52	51.32/21.05	71.71/33.88	81.52
	btsp	$\mathcal{K}_p$	$29.21^{(1)}$	97.22/2.78	97.92/9.72	86.53	$36.31^{(3)}$	41.67/60.42	53.47/70.14	95.45	$27.17^{(2)}$	53.47/16.67	76.39/33.33	91.33
		∅	**$32.93^{(3)}$**	**97.92/9.03**	**98.61/16.67**	96.31	$37.01^{(2)}$	**44.44/61.81**	57.64/73.61	**37.75**	**$28.82^{(2)}$**	**59.72/17.36**	77.78/28.47	91.53
	pbatch100	$\mathcal{K}_p$	$23.91^{(3)}$	93.06/1.39	95.83/9.03	70.81	$36.26^{(2)}$	41.67/59.03	53.47/69.44	97.76	$26.98^{(2)}$	49.31/20.14	**73.61/37.50**	91.18
		∅	$28.92^{(3)}$	87.50/3.47	90.97/8.33	93.25	$37.01^{(2)}$	**44.44/61.81**	55.56/72.92	98.07	$26.10^{(2)}$	50.00/19.44	72.92/37.50	91.18
	baseline, seq B_1		**32.07**	**96.53/4.86**	**97.22/12.50**	92.76	**30.39**	**50.69/35.42**	**63.89/56.94**	97.37	28.05	52.08/21.53	**69.44/36.81**	92.13
	btsp, seq B_1	$\mathcal{K}_p$	$24.67^{(1)}$	70.83/6.25	84.72/15.97	85.50	$16.34^{(1)}$	63.89/15.28	79.86/29.86	66.66	$29.23^{(2)}$	45.83/26.39	64.58/37.50	98.21
		∅	$23.51^{(1)}$	72.22/4.86	81.94/20.14	89.48	$28.57^{(1)}$	55.56/24.31	67.36/40.97	97.95	**$29.83^{(9)}$**	**59.65/15.79**	70.18/35.09	89.67
Reversibility	baseline		23.00	**95.10/1.40**	95.80/3.50	83.55	29.86	17.48/53.85	35.66/64.34	96.29	23.67	93.71/1.40	97.20/4.20	91.25
	single_prop		19.81	91.78/2.05	**95.89/6.85**	**59.84**	30.04	**14.52/61.29**	**35.48/70.97**	87.14	21.05	93.73/0.66	98.35/3.30	80.41
	btsp	$\mathcal{K}_p$	$28.77^{(1)}$	93.71/0.70	94.41/4.90	89.08	$31.73^{(3)}$	12.59/62.94	28.67/72.73	96.96	$24.49^{(3)}$	92.31/0.70	95.80/3.50	92.30
		∅	**$32.43^{(3)}$**	88.81/2.80	93.71/5.59	97.97	**$32.90^{(4)}$**	14.69/60.84	30.07/67.83	99.71	**$24.83^{(3)}$**	93.71/0.70	96.50/4.20	93.11
	pbatch100	$\mathcal{K}_p$	$28.98^{(1)}$	91.61/0.70	93.71/8.39	93.03	$31.63^{(1)}$	13.29/59.44	32.17/69.93	97.93	$24.10^{(2)}$	93.71/2.10	**97.20/5.59**	92.86
		∅	$30.10^{(1)}$	93.01/1.40	94.41/6.29	96.60	$31.34^{(1)}$	15.38/58.04	34.27/69.93	97.93	$23.77^{(2)}$	**94.41/2.10**	**97.20/5.59**	92.76
	baseline, seq B_1		20.25	73.43/9.09	77.62/15.38	85.41	**28.38**	**16.08/51.05**	**33.57/65.03**	98.58	3.07	22.38/41.26	33.57/55.94	**11.73**
	btsp, seq B_1	$\mathcal{K}_p$	**$26.59^{(1)}$**	93.01/0.00	**96.50/4.20**	86.33	**$28.94^{(1)}$**	16.78/48.95	28.67/65.73	88.67	$2.46^{(1)}$	**21.68/53.15**	**31.47/69.93**	**15.62**
		∅	$18.58^{(1)}$	**92.31/2.80**	93.01/6.99	89.18	$4.65^{(1)}$	12.59/38.46	37.76/58.74	**44.06**	**$4.97^{(3)}$**	64.34/7.69	76.22/23.78	**50.07**

Table 2: **Geo**. Property-compliance rate (PCR) and binary property evaluation (BPE) for symmetry and reversibility on Geo. Each cell is reported from the run with the highest PCR, indexed by (k) when multiple runs exist. PCR and BPE (+/−). Port. denotes portability and Eff. injection success. The best value within each group and model is boldfaced and marked by PCR, BPE, or Port.; Eff. < 60 is marked in red. Rows are grouped by injection regime, with the corresponding Baseline shown first.

symmetry), the exception being Llama3-70B symmetry with discard ($94.61 \rightarrow 37.75$), so the compliance gains do not rest on fewer learned facts.$^{(\star)}$ This comes at a cost in specificity on B_1, which decreases in multiple settings (e.g. $90.31 \rightarrow 59.47$ for Llama3-8B symmetry), while paraphrase generalisation improves for Llama3-8B ($62.36 \rightarrow 88.08$; App. Table 7).

Relation to baseline compliance. The magnitude of the improvement is inversely related to the compliance of the unsupervised baseline. On symmetry, it is largest for Llama3-8B, whose baseline PCR is the lowest, and smallest for Llama3-70B, whose baseline is the highest (34.78). The Wikidata corpus (App. Table 8) follows the same trend: its baseline PCR is 30 to 50 points higher than on Geo, and the best Batch point improves on Base in every cell by a smaller margin of 0.3 to 2.2 points (e.g. symmetry on Llama3-8B, $59.14 \rightarrow 61.29$; reversibility on Llama3-70B, $77.34 \rightarrow 79.08$). Baseline compliance does not account for all differences, however: on reversibility, Qwen3-32B and Llama3-8B start from similar baselines (23.67 and 23.00), yet the gain is $+5\%$ for the former against $+41\%$ for the latter, indicating that the difficulty of acquiring a property also depends on the model.

RQ1: property supervision transfers to unseen entities. Property-instantiating edits on one set of entities raise compliance on the counterparts of new facts about entities absent from supervision, by up to $+55\%$ relative. Because no evaluated entity occurs in a supervised pair, the gain cannot stem from memorised pairs: the edits change how the model handles the relation beyond the facts they write. The gain is largest for the model with the lowest baseline and smaller on Wikidata, where baselines are higher.

6.2 RQ2: Supervision scale and injection mode

Effect of supervision scale. Figure 2 varies the amount of supervision by changing the number of examples used to teach the property. For every model, the best point is reached between n=10 and n=150 pairs (superscripts in Table 2), after which compliance saturates and then gradually decreases: mildly for Llama3-70B, which stays close to its baseline, and more strongly for Llama3-8B with keep and for Qwen3-32B beyond 150 pairs. The amount of supervision is thus a parameter to calibrate, with a favourable range at small n.

Effect of preserving property keys. The keep and discard settings share the same θ_1 (Algorithm 1, l. 6), the same B_1 and the same order, so any separation between them is due to the constraint K_p alone. The two start close and separate as n grows, with discard above keep, most clearly for Llama3-8B, whose discard setting also peaks later (k=3 against k=1 on both properties). Efficacy follows the same ordering in different Geo settings (e.g. 86.53 against 96.31 for Llama3-8B symme-

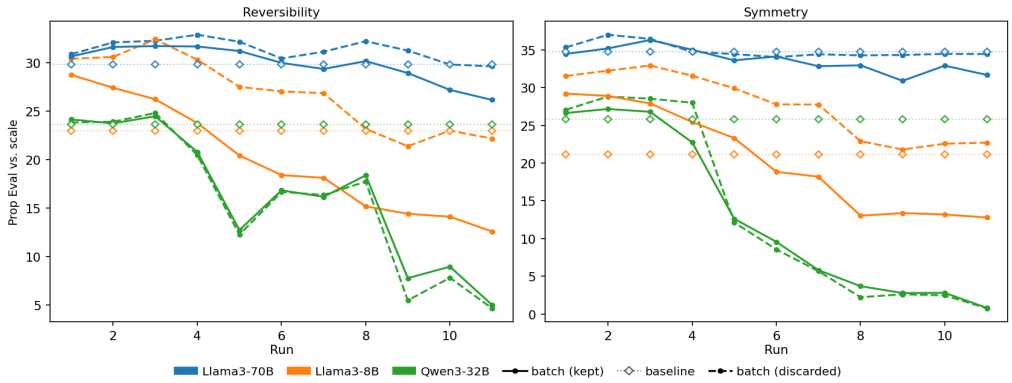


Figure 2: Compliance against the number of supervised pairs n, with $\mathcal{K}_p$ preserved or discarded at the phase boundary. Horizontal lines is the matched baseline. Scales described in App. 3, Wikidata results in App. B.6.

try): under keep, the new fact itself is harder to write, as expected from a more constrained update. Preserving the property keys thus limits peak compliance, while it better preserves compliance under sequential editing (App. C).

Effect of the injection mode. Solving the property phase in one pass is not necessary: Chunk, which splits it into successive batches of 100 pairs, stays above Base in all Geo settings and remains within 1 to 3 points of Batch, falling behind by 4 to 5 points only for Llama3-8B on symmetry, where its efficacy with keep also drops to 70.81. Writing the pairs in the same update as the new fact is different: Joint exceeds Base in only a minority of Geo settings and falls below it for Llama3-8B and Qwen3-32B on both properties (e.g. reversibility on Llama3-8B, $23.00 \rightarrow 19.81$). It also lowers efficacy in all 6 Geo settings, most strongly for Llama3-8B (50.23 and 59.84), indicating that new facts are less often acquired when they compete with the property pairs within a single solve.

RQ2: A limited amount of property supervision, applied before fact injection, is sufficient to obtain the gain. On Geo, the best compliance is reached with 10 to 150 supervised pairs, and larger amounts reduce it, sharply for Qwen3-32B. Splitting the property phase into batches preserves the gain, whereas writing the pairs in the same update as the new fact is unreliable: it falls below the baseline for Llama3-8B and both Qwen3 models and lowers efficacy, although it helps Llama3-70B on symmetry. Preserving the property keys lowers peak compliance on Geo but, for Llama3-8B, makes it more robust to subsequent edits.

6.3 RQ3: Consistency across evaluation formats

Compliance beyond the completion format. Figure 3 evaluates the same counterparts as binary judgements in both polarities: agreement with PCR shows that a result is not an effect of the answer format, and disagreement shows where such an effect is present. At baseline, Llama3-8B separates the positive from the negative question almost perfectly (symmetry 98.61/1.39, reversibility 95.10/1.40), and confirms the edited fact itself just as reliably (Port. 99.31/2.78), yet its PCR is only 21.18 and 23.00: the relation is recognised but rarely completed as expected. Injection narrows this gap from the PCR side but raises BPE^- on symmetry from 1.39 to 9.03, and Port.^- from 2.78 to 16.67, so part of the gain reflects a broader tendency to affirm the relation. Port. also shows when binary judgements can be trusted at all. For Llama3-70B on Geo reversibility, they fail already on the edited fact, which it affirms less often than its opposite (Port. 35.66/64.34); its judgements on counterparts mirror this inversion (BPE 17.48/53.85). Its binary answers are thus consistent with each other but do not register the edit, so its BPE reflects the answer format rather than property compliance. On Wikidata, the same model's affirmation bias (79.86/65.28) is reduced by injection (82.64/47.92).

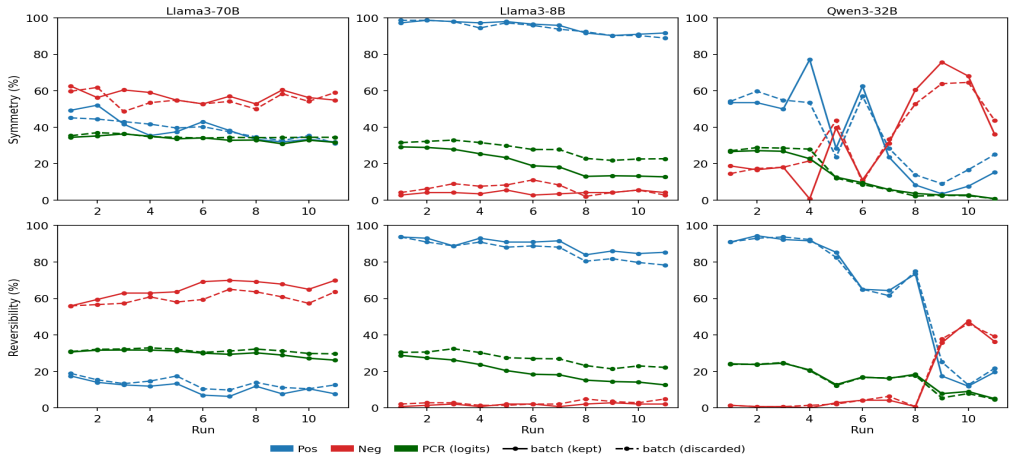


Figure 3: Binary property evaluation (BPE) positive/negative proportions under logit-based scoring on Geo. BPE+., BPE-., and PCR denote the positive proportion, negative proportion, and property-compliance rate, respectively, with PCR shown on the same axes in each cell. Line styles and markers indicate the K_p/curve-mode setting (see legend) [Wikidata in App 5].

RQ3: completion and binary judgements of a property can disagree. Llama3-8B reliably affirms counterparts it rarely completes, while Llama3-70B affirms reversible relations in the edited direction more often than in the inverse one, a pattern consistent with treating them as symmetric. Property learning reconciles the two formats for some models and widens the gap for others. Reporting compliance in a single format would therefore hide part of what editing does to a property.

7 Conclusion and Perspectives

This work extends knowledge editing from isolated facts to the relational properties that govern them. By separating property learning from fact injection and evaluating on unseen entities, PropEdit-Bench and its property-aware metrics provide a controlled environment for probing property abstraction, free from memorisation and domain content. In this environment, a small number of property-instantiating edits, written before a new fact, raise compliance on its unseen counterpart in nearly every model, property and domain (up to $+55\%$) and ease the acquisition of the fact itself, at a cost in specificity; completion and binary judgements, however, can disagree. Editing thus conveys relational structure only partially, and not as a by-product of factual correction: the structure behind knowledge must become an explicit target of editing.

Perspectives. Our findings open three directions, each building on the present protocol and data. **(1) Structure-aware editing.** Injected properties raise compliance but reduce specificity and remain sensitive to later edits (App. C). Since the key of a counterpart is never written (§5.2), new editing objectives that couple each fact with its counterpart (Ma et al., 2023) or constrain how a relation is decoded (Hernandez et al., 2024; Kim et al., 2025) are needed, and our protocol provides a clean environment to evaluate them on disjoint counterparts. **(2) Multi-property learning.** Geo combines symmetric, reversible and transitive relations over the same locations, patterns that relational models can capture jointly (Sun et al., 2019). Mixing pairs of several properties in the property phase, once splits are made collision-free across properties, would show whether properties reinforce one another or compete for the same update (Li et al., 2024; Nishi et al., 2025). **(3) Cross-domain transfer.** Geo and Wikidata share symmetry and reversibility under different surface content. Learning a property in one domain and evaluating it in the other requires aligning properties across relations that share no surface form (Hernandez et al., 2024), but would extend disjointness from entities to domains and provide the strongest probe of whether a model abstracts the property itself (Kim et al., 2025).

REFERENCES

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.2, knowledge manipulation. In *International Conference on Learning Representations*, 2025.

Lukas Berglund, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. The reversal curse: LLMs trained on "A is B" fail to learn "B is A". In *International Conference on Learning Representations*, 2024.

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. Evaluating the ripple effects of knowledge editing in language models. *Transactions of the Association for Computational Linguistics*, 12:283–298, 2024.

Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6491–6506, 2021.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031, 2021.

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Jie Shi, Xiang Wang, Xiangnan He, and Tat-Seng Chua. AlphaEdit: Null-space constrained knowledge editing for language models. In *International Conference on Learning Representations*, 2025.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5484–5495, 2021.

Olga Golovneva, Zeyuan Allen-Zhu, Jason Weston, and Sainbayar Sukhbaatar. Reverse training to nurse the reversal curse. In *First Conference on Language Modeling*, 2024.

Aaron Grattafiori et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Akshat Gupta, Anurag Rao, and Gopala Anumanchipalli. Model editing at scale leads to gradual and catastrophic forgetting. *arXiv preprint arXiv:2401.07453*, 2024a.

Akshat Gupta, Dev Sajnani, and Gopala Anumanchipalli. A unified framework for model editing. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 15403–15418, 2024b.

Tom Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. Aging with GRACE: Lifelong model editing with discrete key-value adaptors. In *Advances in Neural Information Processing Systems*, volume 36, pp. 47934–47959, 2023.

Peter Hase, Mohit Bansal, Been Kim, and Asma Ghandeharioun. Does localization inform editing? surprising differences in causality-based localization vs. knowledge editing in language models. In *Advances in Neural Information Processing Systems*, 2023.

Evan Hernandez, Arnab Sen Sharma, Tal Haklay, Kevin Meng, Martin Wattenberg, Jacob Andreas, Yonatan Belinkov, and David Bau. Linearity of relation decoding in transformer language models. In *International Conference on Learning Representations*, 2024.

Nikolaos Karalis, Georgios M. Mandilaras, and Manolis Koubarakis. Extending the YAGO2 knowledge graph with precise geospatial knowledge. In *The Semantic Web – ISWC 2019*, volume 11779 of *Lecture Notes in Computer Science*, pp. 181–197. Springer, 2019. doi: 10.1007/978-3-030-30796-7_12.

Dong-Kyum Kim, Minsung Kim, Jea Kwon, Nakyeong Yang, and Meeyoung Cha. Bilinear representation mitigates reversal curse and enables consistent model editing. *arXiv preprint arXiv:2509.21993*, 2025.

Omer Levy, Minjoon Seo, Eunsoo Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning*, pp. 333–342. Association for Computational Linguistics, 2017.

Zhoubo Li, Ningyu Zhang, Yunzhi Yao, Mengru Wang, Xi Chen, and Huajun Chen. Unveiling the pitfalls of knowledge editing for large language models. In *International Conference on Learning Representations*, 2024.

Zhengkai Lin, Zhihang Fu, Kai Liu, Liang Xie, Binbin Lin, Wenxiao Wang, Deng Cai, Yue Wu, and Jieping Ye. Delving into the reversal curse: How far can large language models generalize? In *Advances in Neural Information Processing Systems*, 2024.

Wei Liu, Haomei Xu, Hongkai Liu, Zhiying Deng, Ruixuan Li, Heng Huang, Yee Whye Teh, and Wee Sun Lee. Are we evaluating the edit locality of llm model editing properly?, 2026. URL https://arxiv.org/abs/2601.17343.

Jun-Yu Ma, Jia-Chen Gu, Zhen-Hua Ling, Quan Liu, and Cong Liu. Untying the reversal curse via bidirectional language model editing. *arXiv preprint arXiv:2310.10322*, 2023.

Jun-Yu Ma, Hong Wang, Hao-Xiang Xu, Zhen-Hua Ling, and Jia-Chen Gu. Perturbation-restrained sequential model editing. In *International Conference on Learning Representations*, 2025.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35, pp. 17359–17372, 2022.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. In *International Conference on Learning Representations*, 2023.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. Fast model editing at scale. In *International Conference on Learning Representations*, 2022a.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D. Manning, and Chelsea Finn. Memory-based model editing at scale. In *International Conference on Machine Learning*, 2022b.

Kento Nishi, Rahul Ramesh, Maya Okawa, Mikail Khona, Hidenori Tanaka, and Ekdeep Singh Lubana. Representation shattering in transformers: A synthetic study with knowledge editing. In *International Conference on Machine Learning*, 2025.

Jingcheng Niu, Andrew Liu, Zining Zhu, and Gerald Penn. What does the knowledge neuron thesis have to do with knowledge? In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2405.02421.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. RotatE: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*, 2019.

Peng Wang, Zexi Li, Ningyu Zhang, Ziwen Xu, Yunzhi Yao, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. WISE: Rethinking the knowledge memory for lifelong model editing of large language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 53764–53797, 2024.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. Kepler: A unified model for knowledge embedding and pre-trained language representation, 2020. URL https://arxiv.org/abs/1911.06136.

An Yang et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. Editing large language models: Problems, methods, and opportunities. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 10222–10240, 2023.

Zhangdie Yuan and Andreas Vlachos. Capturing symmetry and antisymmetry in language models through symmetry-aware training objectives. *arXiv preprint arXiv:2504.16312*, 2025.

Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. Can we edit factual knowledge by in-context learning? In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4862–4876, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.296. URL https://aclanthology.org/2023.emnlp-main.296/.

Zexuan Zhong, Zhengxuan Wu, Christopher D. Manning, Christopher Potts, and Danqi Chen. MQuAKE: Assessing knowledge editing in language models via multi-hop questions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 15686–15702, 2023.

A Experimental setup

Models. Llama-3-8B, Llama-3.3-70B, Qwen3-8B and Qwen3-32B.

Corpora. Two corpora per property, one geographic and one from Wikidata kinship, so that a result is not an artefact of a single relation family. Table 3 gives the sizes; $|A_1|$ upper-bounds the scale sweep and $|B_1|$ is the number of independent measurements behind each PCR.

Table 3: Corpora and swept values. $|B_1|$ is subsampled to 144 throughout so that every PCR estimate carries comparable variance across corpora; sampling is balanced across relations rather than proportional to the corpus mix.

Corpus	Property	$\lvert A_1\rvert$	$\lvert B_1\rvert$	Scales n
Geo	symmetry	573	144	10, 50, 100, 150, 200, 250, 300, 350, 400, 450, 573
Geo	reversibility	569	143	10, 50, 100, 150, 200, 250, 300, 350, 400, 450, 569
Wikidata	symmetry	900	600	10, 50, 100, 150, 300, 573
Wikidata	reversibility	900	600	10, 50, 100, 150, 300, 573
Sequential conditions (m_P=1 or m_B=seq): $n \in \{10, 150, 300, \lvert A_1\rvert\}$				

Table 4: Conditions as parameter settings. Rows above the rule are treatments, below are controls. All are Algorithm 1 except Joint. κ is swept (keep/discard) for every row with a property phase; Joint, Pre-edit, and Base$_{m_B}$ have no property-phase/B_1-phase boundary, so κ does not apply. The **Config ID** column gives the matching identifier(s) from Table 5.

Condition	Alg.	m_P	m_B	Config ID	Isolates
Batch	1	n	1	`btsp`	property-batch width
Batch-SeqB	1	n	seq	`btsp_seqB1`	accumulating new facts
Seq-Batch	1	100	1	`pbatch100`	width vs. volume
Seq-SeqB	1	1	seq	`pseq_seqB1`	both phases sequential
Joint	2	—	—	—	staged vs. simultaneous
Pre-edit	—	—	—	—	prior knowledge
Base$_{m_B}$	1	n=0	m_B	`baseline`*	editing without π

Table 5: Experiment configurations implemented in PropEdit-Bench, factorised by property-learning strategy (**PropLearn**, columns) and fact-injection regime (**FactInj**, rows). Each filled cell gives the run identifier. ⋆ run on a reduced scale sweep (fewer, sparser scale points than the rest of the grid) due to the added cost of step-by-step accumulation.

FactInj ↓**PropLearn** →	**None**	**Batch**	**Chunked** (k=100)
Single	`baseline`	`btsp`	
Accumulate	`baseline_seqB1`	`btsp_seqB1`⋆	`pbatch100`

Corpus	Property	Llama3-8B		Llama3-70B		Qwen3-8B		Qwen3-32B	
		Eff. ↑	Gen. ↑	Eff. ↑	Gen. ↑	Eff. ↑	Gen. ↑	Eff. ↑	Gen. ↑
Geo	Symmetry	44.44	34.03	36.44	35.08	23.17	24.19	29.11	28.46
	Reversibility	31.87	31.27	30.95	31.16	17.03	16.74	24.59	24.53
Wikidata	Symmetry	58.82	57.32	69.80	68.24	51.52	49.88	55.61	54.77
	Reversibility	75.46	75.96	75.24	71.99	54.73	54.76	60.89	59.54

Table 6: Unedited models on B_1. Eff. and Gen. measure how much of each corpus the base model already satisfies, i.e. the floor any edited condition must exceed to be informative. PCR and BPE are undefined here, being conditioned on the injected fact having been learned, and Spe. is omitted, being 100 by construction before any edit is applied.

Prop.	Mode	$\mathcal{K}_p$	Llama3-8B			Llama3-70B			Qwen3-32B		
			Gen. ↑	Spe. ↑	L_{topk} ↑/KL ↓	Gen. ↑	Spe. ↑	L_{topk} ↑/KL ↓	Gen. ↑	Spe. ↑	L_{topk} ↑/KL ↓
Symmetry	baseline		62.36	90.31	93.69/0.80	79.53	95.81	97.97/0.08	57.09	92.67	96.75/0.31
	single_prop		45.09	56.84	75.53/10.00	70.77	78.90	94.52/0.69	54.66	65.99	77.18/34.21
	btsp	$\mathcal{K}_p$	75.07	67.89	81.11/7.00	81.10	83.81	93.78/0.72	56.77	83.08	91.28/2.89
		∅	88.08	59.47	74.58/14.06	36.07	80.33	93.47/0.94	56.74	82.86	91.69/2.18
	pbatch100	$\mathcal{K}_p$	62.14	60.25	77.44/9.29	83.21	81.22	93.36/0.86	59.14	68.03	80.81/26.31
		∅	83.37	58.06	76.06/10.64	84.46	80.14	92.86/0.98	58.99	68.81	81.00/26.40
	baseline, seq B_1		85.20	37.08	55.03/71.79	81.47	65.11	86.75/12.74	61.05	83.78	92.36/3.27
	btsp, seq B_1	$\mathcal{K}_p$	77.34	42.47	63.28/30.08	59.93	43.00	81.50/45.65	67.21	59.47	75.17/38.04
		∅	82.19	30.25	52.64/66.33	85.16	42.03	69.06/134.11	80.17	16.56	36.28/244.42
Reversibility	baseline		75.37	86.38	93.06/1.35	86.30	93.57	98.49/0.07	66.09	91.61	94.91/9.69
	single_prop		57.00	58.52	77.78/8.11	76.54	78.52	95.03/0.42	60.98	54.23	65.98/124.40
	btsp	$\mathcal{K}_p$	82.34	65.12	81.82/6.98	82.78	81.09	94.35/0.88	66.65	76.31	87.02/11.84
		∅	93.03	54.32	74.21/17.29	87.70	77.82	93.54/1.30	67.30	76.08	87.19/15.38
	pbatch100	$\mathcal{K}_p$	86.32	64.92	82.18/7.10	86.58	85.90	95.16/0.44	67.63	90.21	95.72/4.60
		∅	90.77	62.55	80.92/8.99	87.36	85.17	95.38/0.54	67.81	90.63	95.92/4.58
	baseline, seq B_1		78.32	24.64	42.18/144.53	87.59	64.39	84.39/28.64	7.78	6.41	10.94/602.25
	btsp, seq B_1	$\mathcal{K}_p$	79.45	28.92	50.18/83.41	76.40	50.91	83.19/8.75	9.74	7.10	13.99/560.04
		∅	86.87	18.27	38.97/155.33	34.23	9.73	31.02/709.57	27.22	11.44	23.78/425.54

Table 7: **Geo**. Generalization (Gen.) and specificity (Spe.) for symmetry and reversibility on Geo. Each cell is reported from the run with the highest PCR, the same run selected in the companion property table, so both describe the same edited model. Gen. and Spe. are measured on B_1, the injected facts. L_{topk}/KL is an auxiliary read on specificity, beyond Spe. itself. Rows are grouped by injection regime, with the corresponding Baseline shown first.

B Full results

Tables here give every combination of condition, model, corpus and property at every scale. Each is generated from the run that produced it; Conventions follow Sec. **??**: efficacy, generalization and specificity on B_1; property compliance on B_2 conditioned on the injected fact having been learned.

B.1 Unedited models

Controls. Pre-edit evaluates θ_0 unmodified and bounds how much of B_2 the model already knows. The **None** column above injects B_1 with no property phase, matched to a B_1 *regime* rather than to a single condition, so three instances cover all treatments.

Sampling. Property pairs and B_1 facts are drawn with equal weight per relation rather than in corpus proportion, so no single relation dominates a measurement, and $|B_1|$ is held fixed across corpora so that estimates carry comparable variance. Corpus sizes are given in App. E.

B.2 Efficacy, generalization and specificity

Efficacy states whether the injected fact was installed at all, and so bounds what any property score computed over the same facts can mean. Generalization and specificity report the cost of the edit elsewhere in the model (Tab. 7).

Prop.	Mode	$\mathcal{K}_p$	Llama3-8B				Llama3-70B				Qwen3-32B			
			PCR ↑	BPE (+/−) ↑	Port. ↑	Eff. ↑	PCR ↑	BPE (+/−) ↑	Port. ↑	Eff. ↑	PCR ↑	BPE (+/−) ↑	Port. ↑	Eff. ↑
Symmetry	baseline		59.14	86.11/6.94	90.28/18.06	97.42	71.52	69.44/57.64	82.64/64.58	98.69	54.84	50.69/12.50	65.28/21.53	98.30
	single_prop		58.46	**87.97/8.23**	92.41/17.30	94.74	70.62	71.49/61.85	**83.53/68.67**	98.36	55.29	**51.91/14.33**	64.97/21.97	97.58
	btsp	$\mathcal{K}_p$	60.14$^{(2)}$	83.33/9.03	86.81/18.75	98.68	72.45$^{(5)}$	**73.61/60.42**	81.25/70.14	99.30	**55.76**$^{(3)}$	56.94/8.33	**68.75/19.44**	98.31
		∅	**61.29**$^{(4)}$	84.03/11.11	**88.19/21.53**	99.22	**72.91**$^{(4)}$	72.22/57.64	83.33/66.67	99.27	55.41$^{(3)}$	56.25/9.03	68.06/19.44	98.35
	pbatch100	$\mathcal{K}_p$	60.14$^{(2)}$	83.33/9.03	86.81/18.75	98.68	72.45$^{(5)}$	**73.61/60.42**	81.25/70.14	99.30	**55.76**$^{(3)}$	56.94/8.33	**68.75/19.44**	98.31
		∅	**61.29**$^{(4)}$	84.03/11.11	**88.19/21.53**	99.22	**72.91**$^{(4)}$	72.22/57.64	83.33/66.67	99.27	55.41$^{(3)}$	56.25/9.03	68.06/19.44	98.35
	baseline, seq B_1		**57.89**	**70.14/10.42**	79.17/21.53	98.76	70.22	77.08/39.58	85.42/52.08	99.37	**54.85**	**54.86/10.42**	**69.44/20.14**	98.23
	btsp, seq B_1	$\mathcal{K}_p$	56.55$^{(1)}$	69.44/9.72	**79.17/22.22**	98.89	**71.69**$^{(1)}$	**78.47/49.31**	**86.11/60.42**	98.98	54.75$^{(2)}$	53.47/9.72	68.75/18.75	99.51
		∅	56.12$^{(1)}$	60.42/11.11	72.92/21.53	98.71	**72.91**	72.22/57.64	83.33/66.67	99.27	54.82$^{(3)}$	52.78/10.42	68.75/19.44	99.68
Reversibility	baseline		67.88	90.97/9.03	**93.06/17.36**	98.27	77.34	79.86/65.28	**90.28/72.92**	99.22	64.12	**75.00/2.78**	86.11/18.06	99.02
	single_prop		66.52	93.81/5.09	97.35/11.06	95.64	78.11	**81.71/63.82**	91.06/71.54	99.10	63.97	72.54/2.39	87.16/18.21	98.57
	btsp	$\mathcal{K}_p$	**69.13**$^{(4)}$	95.14/3.47	97.22/7.64	98.51	**79.08**$^{(6)}$	81.94/51.39	89.58/58.33	99.61	64.38$^{(1)}$	74.31/1.39	88.19/20.83	98.74
		∅	68.40$^{(2)}$	**95.83/6.25**	**97.92/12.50**	99.20	79.00$^{(6)}$	82.64/47.92	90.28/56.25	99.55	**64.40**$^{(1)}$	74.31/2.08	**87.50/21.53**	98.74
	pbatch100	$\mathcal{K}_p$	**69.13**$^{(4)}$	95.14/3.47	97.22/7.64	98.51	**79.08**$^{(6)}$	81.94/51.39	89.58/58.33	99.61	64.38$^{(1)}$	74.31/1.39	88.19/20.83	98.74
		∅	68.40$^{(2)}$	**95.83/6.25**	**97.92/12.50**	99.20	79.00$^{(6)}$	82.64/47.92	90.28/56.25	99.55	**64.40**$^{(1)}$	74.31/2.08	**87.50/21.53**	98.74
	baseline, seq B_1		65.71	87.50/11.11	95.83/21.53	98.38	74.85	**76.39/34.72**	85.42/45.14	99.11	61.22	**73.61/4.86**	88.19/20.14	97.15
	btsp, seq B_1	$\mathcal{K}_p$	**66.10**$^{(1)}$	**93.75/18.75**	96.53/30.56	98.60	**77.42**$^{(1)}$	76.39/34.03	**87.50/44.44**	99.38	**63.81**$^{(2)}$	69.44/5.56	86.11/21.53	99.65
		∅	64.49$^{(1)}$	88.19/21.53	**93.75/36.11**	98.53	74.58$^{(1)}$	65.31/37.76	74.49/46.94	96.50	63.53$^{(1)}$	74.31/3.47	**86.81/25.69**	99.31

Table 8: Property-compliance rate (PCR) and binary property evaluation (BPE) on Wikidata. Each cell is reported from the run with the highest PCR, indexed by (k) when multiple runs exist. RQ denotes PCR and RQJ denotes BPE $(+/-)$, conditioned on successful learning of the injected fact. Port. denotes portability and Eff. injection success. The best value within each group and model is boldfaced and marked by PCR, BPE, or Port.; Eff. $<$ 70 is marked in red. Rows are grouped by injection regime, with the corresponding Baseline shown first.

Prop.	Mode	$\mathcal{K}_p$	Llama3-8B			Llama3-70B			Qwen3-32B		
			Gen. ↑	Spe. ↑	L_{topk} ↑/KL ↓	Gen. ↑	Spe. ↑	L_{topk} ↑/KL ↓	Gen. ↑	Spe. ↑	L_{topk} ↑/KL ↓
Symmetry	baseline		94.28	91.44	96.06/0.45	94.56	94.97	97.50/0.17	82.76	92.61	96.75/0.39
	single_prop		91.95	65.06	85.09/5.81	93.42	80.93	94.15/1.89	81.97	81.25	89.82/3.76
	btsp	$\mathcal{K}_p$	95.75	58.25	80.39/14.52	95.39	71.42	89.53/5.33	82.03	75.89	85.53/8.23
		∅	96.90	44.72	72.81/30.41	95.58	73.33	90.50/5.22	82.16	76.19	85.97/7.94
	pbatch100	$\mathcal{K}_p$	95.75	58.25	80.39/14.52	95.39	71.42	89.53/5.33	82.03	75.89	85.53/8.23
		∅	96.90	44.72	72.81/30.41	95.58	73.33	90.50/5.22	82.16	76.19	85.97/7.94
	baseline, seq B_1		97.04	50.19	73.53/28.56	95.62	66.61	87.17/17.00	82.00	84.75	92.31/2.75
	btsp, seq B_1	$\mathcal{K}_p$	96.24	47.50	72.17/28.43	94.96	66.06	87.97/10.50	84.62	74.00	85.89/8.36
		∅	96.67	43.92	71.17/34.41	85.67	55.96	82.79/31.14	82.74	84.24	72.92
82.89/11.80											
Reversibility	baseline		93.88	91.06	96.17/0.74	94.65	96.92	97.86/0.11	86.00	92.56	96.69/0.37
	single_prop		91.58	55.96	77.62/18.72	94.31	88.89	94.68/0.77	85.23	76.47	86.54/14.11
	btsp	$\mathcal{K}_p$	94.29	41.69	69.08/32.22	95.59	76.78	88.50/5.29	86.14	87.64	93.56/1.54
		∅	96.41	40.19	67.83/33.77	95.91	76.28	88.39/5.49	86.13	88.11	93.97/1.41
	pbatch100	$\mathcal{K}_p$	94.29	41.69	69.08/32.22	95.59	76.78	88.50/5.29	86.14	87.64	93.56/1.54
		∅	96.41	40.19	67.83/33.77	95.91	76.28	88.39/5.49	86.13	88.11	93.97/1.41
	baseline, seq B_1		95.87	51.86	75.39/30.44	95.07	75.28	89.83/6.21	83.62	68.00	77.64/73.60
	btsp, seq B_1	$\mathcal{K}_p$	95.43	51.97	76.33/23.68	95.26	75.06	88.42/11.88	87.82	43.00	61.67/103.09
		∅	96.10	49.89	75.19/28.58	92.55	71.06	84.73/32.28	86.82	74.64	86.78/16.27

Table 9: Generalization (Gen.) and specificity (Spe.) on Wikidata. Each cell is reported from the run with the highest PCR, the same run selected in the companion property table, so both describe the same edited model. Gen. and Spe. are measured on B_1, the injected facts. L_{topk}/KL is an auxiliary read on specificity, beyond Spe. itself. Rows are grouped by injection regime, with the corresponding Baseline shown first.

B.3 Wikidata

Results for Wikidata are shown in (Tab. 9) and for property evaluation and injection (Tab. 8).

B.4 Joint injection

Single_Prop mode represent simultaneous injection of property A1/A2 and single facts B1 without a separate property learning phase.

B.5 Binary property compliance

See Figure 3.

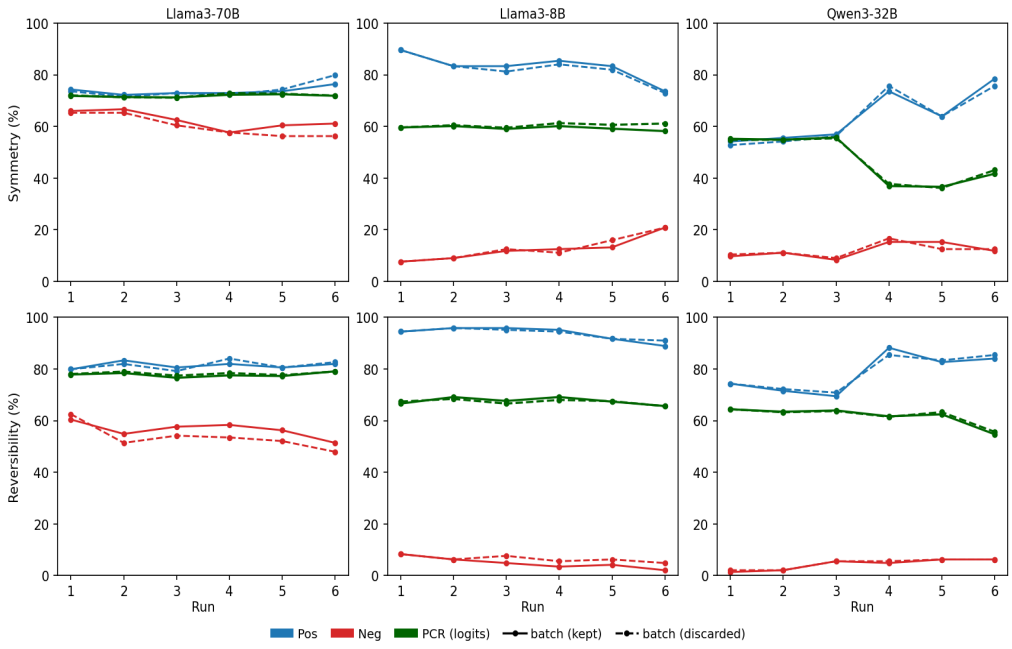


Figure 4: Compliance against the number of supervised pairs n, with $\mathcal{K}_p$ preserved or discarded at the phase boundary between property learning (A1/A2) and B1 fact injection.

B.6 PER-SCALE CURVES

See Figure 5.

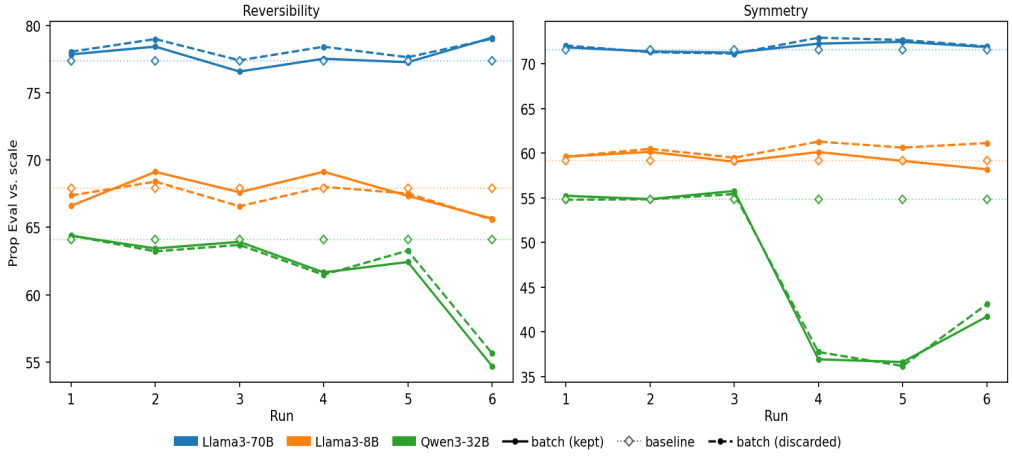


Figure 5: Compliance against the number of supervised pairs n, with $\mathcal{K}_p$ preserved or discarded at the phase boundary. Horizontal lines mark the matched BASE control.

B.7 QWEN3-8B RESULTS

Prop.	Mode	$\mathcal{K}_p$	Wikidata				Geo			
			PCR ↑	BPE (+/−) ↑	Port. ↑	Eff. ↑	PCR ↑	BPE (+/−) ↑	Port. ↑	Eff. ↑
Symmetry	baseline		50.63	20.83/13.19	43.06/21.53	99.25	23.25	79.86/2.08	84.03/18.06	96.73
	single_prop		50.49	19.44/16.44	42.59/26.85	98.91	0.00	44.07/16.10	62.71/34.75	**3.15**
	btsp	$\mathcal{K}_p$	51.01$^{(3)}$	**23.61/20.14**	41.67/30.56	98.97	**26.91**$^{(3)}$	81.25/4.86	86.81/14.58	98.60
		∅	**51.22**$^{(3)}$	**22.92/20.83**	**43.06/30.56**	98.91	26.73$^{(3)}$	**81.94/6.25**	86.81/15.28	98.60
	pbatch100	$\mathcal{K}_p$	51.01$^{(3)}$	**23.61/20.14**	41.67/30.56	98.97	0.04$^{(1)}$	12.50/65.28	**25.00/80.56**	**1.04**
		∅	**51.22**$^{(3)}$	**22.92/20.83**	**43.06/30.56**	98.91	0.11$^{(1)}$	19.44/52.08	31.94/66.67	**3.90**
	baseline, seq B_1		**49.96**	21.53/18.06	43.75/25.69	99.16	**25.65**	75.69/3.47	82.64/18.06	99.14
	btsp, seq B_1	$\mathcal{K}_p$	0.00	26.39/42.36	38.89/54.86	**0.00**	0.00$^{(1)}$	**31.94/49.31**	**47.92/55.56**	**0.00**
		∅	0.00	**29.17/48.61**	**41.67/60.42**	**0.08**	0.00$^{(1)}$	40.28/36.81	50.00/46.53	**0.00**
Reversibility	baseline		**57.13**	48.61/11.81	68.75/22.92	98.89	17.02	79.72/3.50	**86.71/18.18**	96.94
	single_prop		21.82	41.44/23.84	**61.34/44.68**	**48.44**	0.04	70.87/7.09	81.10/22.83	**2.80**
	btsp	$\mathcal{K}_p$	57.02$^{(1)}$	**54.17/13.89**	72.22/23.61	98.97	**20.49**$^{(3)}$	**77.62/6.29**	84.62/13.99	97.41
		∅	56.86$^{(1)}$	**55.56/12.50**	71.53/22.22	98.97	20.26$^{(3)}$	78.32/4.90	86.01/12.59	97.51
	pbatch100	$\mathcal{K}_p$	57.02$^{(1)}$	**54.17/13.89**	72.22/23.61	98.97	18.41$^{(1)}$	76.92/1.40	85.31/13.99	97.23
		∅	56.86$^{(1)}$	**55.56/12.50**	71.53/22.22	98.97	18.61$^{(1)}$	76.92/2.10	85.31/14.69	97.23
	baseline, seq B_1		**42.26**	**40.28/27.78**	**55.56/38.19**	73.61	**0.70**	**51.75/33.57**	58.04/41.26	**4.20**
	btsp, seq B_1	$\mathcal{K}_p$	**58.52**$^{(1)}$	44.00/14.00	66.00/22.00	99.30	**0.70**$^{(1)}$	28.67/51.05	**38.46/65.03**	**4.20**
		∅	56.52$^{(1)}$	43.00/13.55	66.50/21.80	99.30	**0.70**$^{(1)}$	39.86/42.66	47.55/53.85	**4.20**

Table 10: Property-compliance rate (PCR) and binary property evaluation (BPE) on Qwen3-8B, broken out by dataset. Each cell is reported from the run with the highest PCR, indexed by (k) when multiple runs exist. Port. denotes portability and Eff. injection success. The best value within each group and model is boldfaced and marked by PCR, BPE, or Port.; Eff. < 70 is marked in red. Rows are grouped by injection regime, with the corresponding Baseline shown first.

Prop.	Mode	$\mathcal{K}_p$	Wikidata			Geo		
			Gen. ↑	Spe. ↑	L_{topk} ↑/KL ↓	Gen. ↑	Spe. ↑	L_{topk} ↑/KL ↓
Symmetry	baseline		90.77	93.72	97.56/0.16	79.82	93.50	97.61/0.18
	single_prop		90.02	79.85	91.06/4.34	3.52	4.07	14.71/551.17
	btsp	$\mathcal{K}_p$	90.15	73.94	87.47/8.17	82.13	79.06	88.44/4.75
		∅	90.15	74.19	87.50/8.13	82.13	78.03	88.28/4.68
	pbatch100	$\mathcal{K}_p$	90.15	73.94	87.47/8.17	0.71	1.61	5.17/704.84
		∅	90.15	74.19	87.50/8.13	2.53	2.42	7.00/639.75
	baseline, seq B_1		90.40	85.17	92.94/1.77	82.12	82.50	91.47/2.73
	btsp, seq B_1	$\mathcal{K}_p$	0.00	0.19	0.69/703.84	0.00	0.03	0.08/678.15
		∅	0.06	0.06	0.33/619.09	0.00	0.03	0.25/702.75
Reversibility	baseline		90.88	94.36	96.67/6.00	83.07	92.42	95.13/18.04
	single_prop		44.77	34.75	48.69/284.01	2.48	4.35	17.54/521.80
	btsp	$\mathcal{K}_p$	91.48	88.75	94.83/6.55	82.56	77.51	86.27/22.90
		∅	91.35	89.97	94.69/7.18	82.90	77.65	86.74/19.09
	pbatch100	$\mathcal{K}_p$	91.48	88.75	94.83/6.55	81.77	88.84	92.92/16.57
		∅	91.35	89.97	94.69/7.18	81.73	88.81	92.59/18.02
	baseline, seq B_1		66.86	67.11	70.17/155.84	3.44	3.94	4.22/773.05
	btsp, seq B_1	$\mathcal{K}_p$	90.40	85.84	92.24/2.11	3.29	3.78	4.03/753.92
		∅	89.80	84.31	91.23/2.21	3.43	3.80	4.20/799.38

Table 11: Generalization (Gen.) and specificity (Spe.) on Qwen3-8B, broken out by dataset. Each cell is reported from the run with the highest PCR, the same run selected in the companion property table, so both describe the same edited model. Gen. and Spe. are measured on B_1, the injected facts. L_{topk}/KL is an auxiliary read on specificity, beyond Spe. itself. Rows are grouped by injection regime, with the corresponding Baseline shown first.

C PROPERTY PERSISTENCE UNDER SEQUENTIAL FACT INJECTION

This section analyses the rows "seq B_1" of Tables 2 and 8, in which the new facts of B_1 are written one after another and accumulate (m_B=seq), instead of being written on a fresh copy of θ_1. The question is whether compliance acquired in the property phase persists once further facts are edited.

Matched comparison. Sequential editing changes baseline compliance by itself, in both directions (Geo symmetry: $21.18 \rightarrow 32.07$ for Llama3-8B, $34.78 \rightarrow 30.39$ for Llama3-70B). Each supervised row is therefore compared with the Base row of the same regime. In this regime, property supervision exceeds the baseline different cases, against facts edited individually. On Geo symmetry, Llama3-8B reaches 24.67 (keep) and 23.51 (discard), below the sequential baseline of 32.07; the largest remaining margin is Llama3-8B reversibility with keep (26.59 against 20.25).

Role of the property keys. Relative to their values under individual editing, the keep setting loses less than the discard setting. For Llama3-8B on Geo, symmetry decreases by 16% with keep and 29% with discard, and reversibility by 8% and 43%; on Wiki, the same ordering holds for Llama3-8B at smaller magnitudes (4 to 8%). Across both domains, keep stays above the sequential baseline in differents cases. The two settings thus present a trade-off: discard reaches higher compliance when facts are edited individually (§6.2), whereas keep better preserves it as edits accumulate.

A model-specific failure. For Qwen3-32B on Geo reversibility, sequential editing fails to install the new facts (Eff. 11.73 for Base, 15.62 and 50.07 with supervision). The failure is present without property supervision and therefore stems from sequential editing on this model and property.

Summary. Under sequential fact injection, the benefit of property supervision is largely lost against a matched sequential baseline. Preserving the property keys slows this loss without preventing it, so maintaining injected properties under continued editing remains an open problem for current editors.

D Dataset construction

Every instance satisfies three constraints: each subject–relation pair (s, r) occurs once; every fact has positive and negative binary formulations, which rules out one-to-many cases; and objects are diversified so that many (s, r) pairs do not collapse onto the same o.

Within each domain we select relation groups sharing subjects—*topology*, *distance*, *direction* for geography; *sibling*, *spouse*, *father* for kinship—with at least one relation per group symmetric or reversible (the two are mutually exclusive on a relation).

GeoProp-Edit. From $255{,}530$ (s, o) couples we build, per fact: a *cloze* ("Paris is south of"), a *question* form, LLM-generated *paraphrases*[2], *specificity* probes drawn from Wikidata with the same relation but disjoint arguments, and *binary portability* in both polarities. Where several objects share an (s, r) pair we keep one, choosing to maximise object diversity (Alg. 3).

Property splits. Symmetry and reversibility each impose an extra constraint, so their splits are built separately. For symmetry, an object o chosen for (s, r) must not already appear as a subject with r, since both (s, r, o) and (o, r, s) enter the data; for reversibility the same holds for (o, r^{-1}, s). Satisfying this sometimes costs object diversity, which we accept (Algs. 4–5).

Wikidata. We use Wikidata5m (Wang et al., 2020) rather than the live endpoint, for reproducibility and to work offline. The kinship relations are P22 (*father*, reversible with *child*), P26 (*spouse*, symmetric) and P3373 (*sibling*, symmetric). We keep subjects linked to at least one object per relation and diversify objects as above; specificity probes reuse the same relation on disjoint arguments. Sibling chains in the source data can make a symmetric counterpart collide with an existing fact on the same (s, r) pair; we discard the conflicting fact when that occurs. For reversibility, *child* is absent from the source data, so (o, r^{-1}, s) is created for every (s, r, o).

Algorithm 3: Object diversification for subject–relation pairs

Require: Fact groups $\mathcal{G}_{s,r}$
Ensure: One fact per pair (s, r)
$\mathcal{V} \leftarrow \emptyset$ ▷ recently used objects
for all (s, r) **do**
  $P \leftarrow \{o \mid (s, r, o) \in \mathcal{G}_{s,r}\}$
  $C \leftarrow P \setminus \mathcal{V}$
  if $C = \emptyset$ **then**
    $\mathcal{V} \leftarrow \emptyset$
    $C \leftarrow P$
  end if
  Choose (s, r, o) with $o \in C$ at random and retain it
  $\mathcal{V} \leftarrow \mathcal{V} \cup \{o\}$
end for

Algorithm 4: Symmetric pair selection

Require: Fact groups $\mathcal{G}_{s,r}$, symmetric relations $\mathcal{R}_{sym}$
$\mathcal{S} \leftarrow \emptyset$
for all selected facts (s, r, o) **do**
  if $r \in \mathcal{R}_{sym}$ **and** $(s, r) \notin \mathcal{S}$ **then**
    Find o' such that $(s, r, o') \in \mathcal{G}_{s,r}$ and $(o', r) \notin \mathcal{S}$
    if o' exists **then**
      Add (s, r, o') and its counterpart (o', r, s)
      $\mathcal{S} \leftarrow \mathcal{S} \cup \{(s, r), (o', r)\}$
    end if
  end if
end for

Algorithm 5: Reversible pair construction

Require: Fact groups $\mathcal{G}_{s,r}$, inverse relations $\mathcal{R}_{inv}$ with $r \mapsto r^{-1}$
$\mathcal{S}_{rev} \leftarrow \emptyset$
for all selected facts (s, r, o) **do**
  if $r \in \mathcal{R}_{inv}$ **and** $(s, r) \notin \mathcal{S}_{rev}$ **then**
    Find o' in $\mathcal{G}_{s,r}$ such that $(o', r^{-1}) \notin \mathcal{S}_{rev}$
    if o' exists **then**
      Add (s, r, o') and its counterpart (o', r^{-1}, s)
      $\mathcal{S}_{rev} \leftarrow \mathcal{S}_{rev} \cup \{(s, r), (o', r^{-1})\}$
    end if
  end if
end for

Table 12: Comparison of knowledge-editing datasets on relational property support. **Rev** = reverse/inverse queries supported. **Sym**, **Trans** = symmetry and transitivity supported. Datasets marked **(Ours)** are introduced in this work.

Dataset	Reverse	Symmetry	Transitivity
CounterFact (Meng et al., 2022)	✗	✗	✗
ZsRE (Levy et al., 2017)	✗	✗	✗
BAKE (Ma et al., 2023)	✓	✓	✗
MQuAKE (Zhong et al., 2023)	✗	✗	✓
Geo-Fact *(Ours, primary)*	✓	✓	✓
Wikidata subset *(Ours, transfer)*	✓	✓	✓

Property	Config.	Distance		Direction				Topology		Total	Gen.	Spe.	Bin.
		Far	Close	N	E	S	W	Borders	Within				
Normal	*Baseline*	506	506	485	498	489	500	506	506	**3,996**	31,358	7,501	3,996
	Prop. Learning	506	506	485	498	489	500	506	506	**3,996**	31,358	7,501	3,996
Symmetry	*Baseline*	238/238	253/253	-	-	-	-	226/226	-	**717/717**	4,103/4,103	2,189/2,153	717/717
	Prop. Learning	421/55	452/54	-	-	-	-	417/35	-	**1,290/144**	7,358/848	2,886/632	1,290/144
Reversibility	*Baseline*	-	-	175/167	202/168	167/175	168/202	-	-	**712/712**	7,120/7,120	2,219/2,218	712/712
	Prop. Learning	-	-	305/37	337/33	310/32	329/41	-	-	**1,281/143**	12,810/1,430	2,942/634	1,281/143
Transitive	*Baseline*	-	-	-	-	-	-	-	2,200/1,100	**2,200/1,100**	13,200/6,600	10,900/5,474	2,200/1,100
	Prop. Learning	-	-	-	-	-	-	-	2,860/440	**2,860/440**	13,200/2,640	10,900/2,197	2,860/440

Table 13: GeoEdit dataset statistics. Edited facts are reported for each relation, along with supplementary questions covering generalization, specificity, and binary portability; positive and negative counts are equal. The notation "a/b" denotes the counts in the edit set E and test set T, respectively.

Property	Config.	Has Child	Has Father	Sibling	Spouse	Total	Gen.	Spe.	Bin.
Normal	*Baseline*	-	7,507	7,507	7,507	**22,521**	127,619	63,418	22,521
	Prop. Learning	-	7,507	7,507	7,507	**22,521**	127,619	63,418	22,521
Symmetry	*Baseline*	-	-	796/796	704/704	**1,500/1,500**	8,204/8,204	6,947/7,043	1,500/1,500
	Prop. Learning	-	-	1,284/308	1,116/292	**2,400/600**	13,116/3,292	10,562/2,929	2,400/600
Reversibility	*Baseline*	318/1,182	1,182/318	-	-	**1,500/1,500**	9,000/9,000	6,922/6,858	1,500/1,500
	Prop. Learning	1,218/282	1,182/318	-	-	**2,400/600**	14,400/3,600	10,817/2,921	2,400/600
Transitivity	*Baseline*	-	-	12,162/6,081	-	**12,162/6,081**	60,810/30,405	18,244/15,091	12,162/6,081
	Prop. Learning	-	-	15,810/2,433	-	**15,810/2,433**	60,810/12,165	18,244/8,979	15,810/2,433

Table 14: Wikidata (Social).

E Dataset statistics

E.1 Splits

Entities are partitioned into disjoint halves A and B. Each half contributes base facts and their property counterparts under the map π: A_1 and $A_2 = \pi(A_1)$, B_1 and $B_2 = \pi(B_1)$. With E the edit set and T the test set,

$$\textbf{Property Learning: } E = A_1 \cup A_2 \cup B_1, \;\; T = B_2; \qquad \textbf{Baseline: } E = A_1 \cup B_1, \;\; T = B_2. \tag{5}$$

Comparing T across the two isolates the effect of exposing the editor to property instances. Disjointness of A and B makes any gain on B_2 attributable to abstraction rather than memorisation.

CounterFact schema. We cast the data into the CounterFact format for compatibility with existing pipelines: the cloze becomes the requested rewrite, paraphrases the paraphrase prompts, specificity clozes and questions the neighborhood and attribute prompts, and binary portability the generation prompts. Three bookkeeping fields track relationships: `idx` identifies the (s, r) pair, `pair_idx` the linked counterpart under π, and `prop_idx` the derived property question.

F Metrics

For a fact (s, r, o) we build a query x from (s, r) whose expected answer is the edited object o^*, tokenized as $y = (y^{(1)}, \ldots, y^{(n)})$ over vocabulary V. f_{θ_0} is the model before editing and f_{θ^*} after; $P_f(v \mid x)$ and $\ell_f(v \mid x)$ are the probability and logit f assigns to v as continuation of x.

Efficacy and generalization. Under teacher forcing, step j scores whether the gold token is the argmax given the correct prefix:

$$m(x, y) = \frac{1}{n} \sum_{j=1}^{n} \mathbb{1}\left[y^{(j)} = \arg\max_{v \in V} P_{f_{\theta^*}}\left(v \mid x \| y^{(1)} \cdots y^{(j-1)}\right) \right]. \tag{6}$$

Efficacy applies Eq. equation 6 to the rewrite prompt, generalization to each paraphrase. An instance evaluated on a query set $\mathcal{Q}$ scores $M = |\mathcal{Q}|^{-1} \sum_k m(x_k, y_k)$.

[2] Verified manually

Component	Content
Example GeoBench	
Editing Knowledge	Prenton → west → Prescot
Generalization	Prenton lies west of
Locality	Gramsh sits west of → Patos
Binary Portability Positive	Is Prenton located west of Prescot? → Yes
Binary Portability Negative	Is Prenton located east of Prescot? → No
Example GeoBench	
Editing Knowledge	Sowerby Bridge → close → Kingstone
Generalization	Sowerby Bridge is near
Locality	Kiryat Yam lies close to → Baqa-Jatt
Binary Portability Positive	Is Sowerby Bridge located close to Kingstone? → Yes
Binary Portability Negative	Is Sowerby Bridge located far from Kingstone? → No
Example Wikidata (Social)	
Editing Knowledge	Helena → has father → Constantine the Great
Generalization	Helena is the offspring of
Locality	Irakli Bagration of Mukhrani is the offspring of → Constantine IV, Prince of Mukhrani
Binary Portability Positive	Is Helena the child of Constantine the Great? → Yes
Binary Portability Negative	Is Helena not the child of Constantine the Great? → No
Example Wikidata (Social)	
Editing Knowledge	Macaulay Culkin → spouse → Rachel Miner
Generalization	Macaulay Culkin is the partner of
Locality	André Bjerke is the husband/wife of → Henny Moan
Binary Portability Positive	Is Macaulay Culkin the spouse of Rachel Miner? → Yes
Binary Portability Negative	Is Macaulay Culkin not the spouse of Rachel Miner? → No

Table 15: Representative examples from ours datasets. Each instance includes editing knowledge, forward prompts for generalization and locality, and binary portability prompts for RQJ evaluation.

Binary portability. The binary question is run once without teacher forcing. With $V_{\text{yes}}, V_{\text{no}} \subset V$ the tokens decoding to "yes"/"no", the answer is YES iff $|V_{\text{yes}}|^{-1}\sum_{v\in V_{\text{yes}}} \ell_{f_{\theta^*}}(v \mid x) > |V_{\text{no}}|^{-1}\sum_{v\in V_{\text{no}}} \ell_{f_{\theta^*}}(v \mid x)$.

Specificity. Editing must not change generations elsewhere. We greedily decode k tokens from both models (fixed seed) and require $\text{Greedy}_k(f_{\theta_0}, x) = \text{Greedy}_k(f_{\theta^*}, x)$. We additionally report $D_{KL}\big(P_{f_{\theta_0}}(\cdot \mid x) \| P_{f_{\theta^*}}(\cdot \mid x)\big)$ and the top-k overlap $L_{\text{topk}}(x) = |S_k^{\theta_0}(x) \cap S_k^{\theta^*}(x)|/k$ following (Liu et al., 2026).

Thresholding. A raw $m(x, y) \in [0, 1]$ does not itself say whether a prediction is correct, and a single cutoff is arbitrary, so we average the indicator over the grid $\mathcal{T} = \{0.1, \ldots, 1.0\}$: $\bar{m}(x, y) = |\mathcal{T}|^{-1}\sum_{th\in\mathcal{T}} \mathbb{1}[m(x, y) > th]$, and M^{th} is its mean over $\mathcal{Q}$.

Property compliance. Let $\mathcal{P} = \{(x_1^{(t)}, y_1^{(t)}, x_2^{(t)}, y_2^{(t)})\}_{t=1}^N$ pair each edited question x_1 with its counterpart $x_2 = \pi(x_1)$ in the test set, and let $\mu \in \{\text{Eff}, \text{Bin}^+, \text{Bin}^-\}$ be a base metric with scores $\mu_1^{(t)}, \mu_2^{(t)}$. The soft version is the mean product, high only when both sides are answered:

$$RQ_\mu = \frac{1}{N}\sum_{t=1}^{N} \mu_1^{(t)}\,\mu_2^{(t)}. \tag{7}$$

The threshold version conditions on the edited question having been learned. With $\mathcal{C}_\mu^{th} = \{\, t : \mu_1^{(t)} > th \,\}$,

$$RQ_\mu^{th} = \frac{1}{|\mathcal{T}|}\sum_{th\in\mathcal{T}} \frac{1}{|\mathcal{C}_\mu^{th}|}\sum_{t\in\mathcal{C}_\mu^{th}} \mathbb{1}\big[\mu_2^{(t)} > th\big]. \tag{8}$$

$\mu = \text{Eff}$ gives RQ^{th}, our headline measure; the binary variants give RQJ_{pos}^{th} and RQJ_{neg}^{th}, and requiring both polarities on both sides gives RQJ_{const}^{th}. We report $|\mathcal{C}^{th}|$ alongside RQ^{th}, since it is

the effective sample behind each score and varies across conditions.RQ **denotes PCR and** RQJ **denotes BPE (+/−)**

F.1 Application: Property Evaluation

Prop.	Mode	$\mathcal{K}_p$	Llama3-8B			Llama3-70B			Qwen3-8B			Qwen3-32B		
			Logits ↑	Soft th=0.5 ↑	Mean th ↑	Logits ↑	Soft th=0.5 ↑	Mean th ↑	Logits ↑	Soft th=0.5 ↑	Mean th ↑	Logits ↑	Soft th=0.5 ↑	Mean th ↑
Symmetry	baseline		21.18	25.69	24.72	34.78	41.67	34.44	23.25	24.31	22.01	25.78	30.56	26.94
	single_prop		17.99	25.81	24.13	38.57	58.06	40.48	0.00	0.00	0.00	24.52	29.28	26.64
	btsp	$\mathcal{K}_p$	29.21$^{(1)}$	33.33$^{(1)}$	29.65$^{(1)}$	36.31$^{(3)}$	47.22$^{(3)}$	35.76$^{(3)}$	26.91$^{(3)}$	26.39$^{(3)}$	25.28$^{(3)}$	27.17$^{(2)}$	31.25$^{(2)}$	27.43$^{(2)}$
		∅	32.93$^{(3)}$	38.19$^{(3)}$	31.88$^{(3)}$	37.01$^{(2)}$	47.22$^{(2)}$	35.69$^{(2)}$	26.73$^{(3)}$	27.08$^{(3)}$	25.14$^{(3)}$	28.82$^{(2)}$	32.64$^{(2)}$	28.68$^{(2)}$
	pbatch100	$\mathcal{K}_p$	23.91$^{(3)}$	29.86$^{(3)}$	26.88$^{(3)}$	36.26$^{(2)}$	46.53$^{(2)}$	35.07$^{(2)}$	0.04$^{(1)}$	0.00$^{(1)}$	0.07$^{(1)}$	26.98$^{(2)}$	31.94$^{(2)}$	27.36$^{(2)}$
		∅	28.92$^{(3)}$	34.72$^{(3)}$	28.13$^{(3)}$	37.01$^{(2)}$	47.22$^{(2)}$	35.69$^{(2)}$	0.11$^{(1)}$	0.00$^{(1)}$	0.28$^{(1)}$	26.10$^{(2)}$	30.56$^{(2)}$	26.53$^{(2)}$
	baseline, seq B_1		32.07	43.06	31.94	30.39	36.81	29.03	25.65	26.39	24.24	28.05	31.25	27.85
	btsp, seq B_1	$\mathcal{K}_p$	24.67$^{(1)}$	31.94$^{(1)}$	26.46$^{(1)}$	2.59$^{(1)}$	4.84$^{(1)}$	5.99$^{(1)}$	0.00$^{(1)}$	0.00$^{(1)}$	0.00$^{(1)}$	29.23$^{(2)}$	31.25$^{(2)}$	27.92$^{(2)}$
		∅	23.51$^{(1)}$	27.08$^{(1)}$	23.82$^{(1)}$	28.57$^{(1)}$	35.42$^{(1)}$	27.36$^{(1)}$	0.00$^{(1)}$	0.00$^{(1)}$	0.00$^{(1)}$	29.83$^{(9)}$	29.82$^{(9)}$	27.89$^{(9)}$
Reversibility	baseline		23.00	25.17	23.92	29.86	34.97	28.67	17.02	12.59	15.59	23.67	25.87	22.73
	single_prop		19.81	28.08	24.93	30.04	41.94	29.84	0.04	0.00	0.08	21.05	23.76	21.52
	btsp	$\mathcal{K}_p$	28.77$^{(1)}$	34.97$^{(1)}$	29.23$^{(1)}$	31.73$^{(3)}$	38.46$^{(3)}$	30.63$^{(3)}$	20.49$^{(3)}$	18.18$^{(3)}$	18.95$^{(3)}$	24.49$^{(3)}$	25.17$^{(3)}$	23.50$^{(3)}$
		∅	32.43$^{(3)}$	36.36$^{(3)}$	30.35$^{(3)}$	32.90$^{(4)}$	40.56$^{(4)}$	31.12$^{(4)}$	20.26$^{(3)}$	18.18$^{(3)}$	18.74$^{(3)}$	24.83$^{(3)}$	25.87$^{(3)}$	23.64$^{(3)}$
	pbatch100	$\mathcal{K}_p$	28.98$^{(1)}$	34.97$^{(1)}$	28.60$^{(1)}$	31.63$^{(1)}$	36.36$^{(1)}$	29.93$^{(1)}$	18.41$^{(1)}$	15.38$^{(1)}$	16.85$^{(1)}$	24.10$^{(2)}$	26.57$^{(2)}$	23.08$^{(2)}$
		∅	30.10$^{(1)}$	35.66$^{(1)}$	28.88$^{(1)}$	31.34$^{(1)}$	35.66$^{(1)}$	29.65$^{(1)}$	18.61$^{(1)}$	15.38$^{(1)}$	17.13$^{(1)}$	23.77$^{(2)}$	26.57$^{(2)}$	22.66$^{(2)}$
	baseline, seq B_1		20.25	22.38	21.26	28.38	32.17	26.78	0.70	1.40	0.70	3.07	4.90	3.36
	btsp, seq B_1	$\mathcal{K}_p$	26.59$^{(1)}$	30.07$^{(1)}$	28.11$^{(1)}$	**28.94**$^{(1)}$	**37.04**$^{(1)}$	**30.43**$^{(1)}$	1.40$^{(1)}$	0.70$^{(1)}$	2.46$^{(1)}$	4.20$^{(1)}$	4.61$^{(1)}$	2.87$^{(1)}$
		∅	18.58$^{(1)}$	19.58$^{(1)}$	18.25$^{(1)}$	4.65$^{(1)}$	8.45$^{(1)}$	9.72$^{(1)}$	0.70$^{(1)}$	1.40$^{(1)}$	0.70$^{(1)}$	4.97$^{(3)}$	4.20$^{(3)}$	5.94$^{(3)}$

Table 16: Property-compliance rate (PCR) across scoring variants on Geo. Each cell is reported from the run with the highest PCR, indexed by (k) when multiple runs exist, using the same run column across all three variants. Logits is the continuous logits-based PCR used elsewhere in this paper; Soft th$=0.5$ and Mean th threshold the same per-fact score at 0.5 and 1 respectively before averaging. Rows are grouped by injection regime, with the corresponding Baseline shown first.

G Transitivity Property Learning

Transitivity poses a structural editing challenge that is not captured by treating each fact as an independent training item. For a relation r, a transitive chain is represented by

$$P_1 = (x, r, y), \qquad P_2 = (y, r, z), \qquad P_1, P_2 \Rightarrow C = (x, r, z).$$

The held-out consequence C is supported jointly by two premises, so one property target is associated with multiple edit records. Consequently, a transitivity *B1 item* is a pair of premises rather than a single record. Property learning must preserve this group structure throughout batching and injection: the two premises are edited as a unit, while the corresponding conclusion is evaluated as the single downstream consequence. Grouping the data by the shared property index yields groups of size two when the conclusion is held out and size three when the complete chain is present, preventing a premise from being detached from the property it jointly instantiates.

G.1 Binary Formulation for Property Learning

The second challenge is the mismatch between completion-style supervision and the logical form of a transitivity conclusion. Each premise admits both affirmative and negated probes, whereas a conclusion probe has a single target polarity. Under completion supervision, the conclusion target is structurally incompatible with the prompt construction: the desired answer is always the opposite polarity from the record's completion target. We therefore formulate transitivity itself as a binary decision problem,

$$f_\theta(x, r, z) \in \{\text{Yes}, \text{No}\},$$

so that the same logical quantity is represented consistently during training, property injection, and evaluation. This is a change to the editing representation, not a post-hoc evaluation transformation. Premises are converted to same-polarity yes/no questions and conclusions retain their corresponding binary probes; paraphrases are generated from the source relation templates with matched polarity. The resulting B1 and test examples thus share an identical answer space while preserving the original transitivity linkage.

G.2 Injection and Evaluation

For controlled comparison, the transitivity B1 pool contains 288 premise records organized into 144 complete chain pairs, so one injection step corresponds to one full pair rather than one record. The property pool contains 550 relation-stratified groups, and property-learning scales are evaluated over increasing numbers of these groups. The baseline uses the same B1 draw without a property-learning phase, while the treatment learns the property from grouped premises before testing the held-out conclusion. In particular, rewrite efficacy on B1 is the mean correctness of the two premises in a pair, and test efficacy is the correctness of its single held-out conclusion. Generation uses six same-polarity paraphrases on both sides; specificity and the neighborhood-based metrics retain the same definition used for symmetry and reversibility. Binary portability is polarity-aligned: a transitivity conclusion contains only its realized polarity, so the premise-side contribution is aggregated only for the matching positive or negative probe. This preserves a one-to-one comparison between the learned property signal and the held-out conclusion rather than duplicating or mixing polarity-specific evidence.

Compared with symmetry and reversibility, the key difference is therefore not the downstream metric family but the unit of property learning. Symmetry/reversibility can be processed record-wise, whereas transitivity requires pair-aware editing and binary supervision from the outset. This formulation exposes whether the model learns the compositional implication encoded by two premises, rather than whether it can merely reproduce an open-ended completion format.

G.3 Results

Table 17 shows that property learning also transfers to transitivity. When new premise pairs are edited individually, Batch raises PCR over Base for all three models and both values of κ, from 64.24 to 73.61 on Llama3-8B ($+15\%$), from 63.52 to 70.14 on Llama3-70B ($+10\%$) and from 57.29 to 62.15 on Qwen3-32B ($+8\%$). Efficacy exceeds 94% in every row, so these estimates rest on nearly all injected premises; Chunk retains smaller gains in all six settings. Two patterns differ from symmetry and reversibility. First, Joint also exceeds Base for all three models and gives the best PCR for Qwen3-32B (62.86), whereas it was unreliable for the other properties (§6.2). Second, keep matches or exceeds discard (Llama3-8B: 73.61 against 68.40), and under sequential injection, where editing alone already raises baseline compliance (Llama3-8B: $64.24 \rightarrow 91.67$), property learning remains above the sequential baseline in five of six settings, the exception being Llama3-8B with discard (76.00 against 91.67). Binary judgements again diverge from PCR: for Llama3-8B, supervision lowers BPE^{+} ($68.24 \rightarrow 52.03$) and raises BPE^{-} ($22.86 \rightarrow 39.29$) while PCR rises, and Qwen3-32B affirms both polarities at similar rates already on the premises themselves (Port. $91.22/87.86$), so its binary answers carry little information. Finally, as for the other properties, property learning improves generalisation on B_1 for all models but reduces specificity, most strongly for Llama3-8B ($86.57 \rightarrow 51.57$; Table 18).

Prop.	Mode	$\mathcal{K}_p$	Llama3-8B				Llama3-70B				Qwen3-32B			
			PCR ↑	BPE (+/−) ↑	Port. ↑	Eff. ↑	PCR ↑	BPE (+/−) ↑	Port. ↑	Eff. ↑	PCR ↑	BPE (+/−) ↑	Port. ↑	Eff. ↑
Transitivity	baseline		64.24	68.24/22.86	75.00/45.00	94.44	63.52	38.05/20.32	49.05/49.26	97.57	57.29	70.95/65.00	91.22/87.86	98.61
	single_prop		66.32	62.50/25.71	70.27/48.57	96.18	66.67	40.00/21.43	50.00/50.00	97.92	**62.86**	**68.75/81.58**	**90.62/92.11**	100.00
	btsp	$\mathcal{K}_p$	**73.61**$^{(4)}$	56.08/31.43	65.54/54.29	97.22	69.91	68.27/22.32	77.88/49.11	99.54	62.15$^{(2)}$	70.27/74.29	91.22/90.00	98.61
		∅	68.40$^{(4)}$	**52.03/39.29**	**61.49/60.00**	94.79	**70.14**	**75.00/23.57**	**84.46/45.00**	100.00	61.46	69.59/72.86	91.89/89.29	98.26
	pbatch100	$\mathcal{K}_p$	66.02	53.17/30.00	61.11/53.08	94.92	68.40	75.68/18.57	82.43/45.71	99.65	59.03	74.32/67.14	92.57/87.86	98.61
		∅	65.28	52.38/30.71	60.32/53.85	94.58	69.10	75.68/18.57	82.43/45.71	99.65	60.42	72.97/69.29	92.57/88.57	98.26
	baseline, seq B_1		91.67	**92.57/62.14**	**97.30/92.86**	100.00	76.83	85.34/18.33	92.24/58.33	100.00	61.25	68.37/70.59	92.86/93.14	98.00
	btsp, seq B_1	$\mathcal{K}_p$	**92.96**$^{(1)}$	87.50/58.57	94.44/82.86	100.00	78.45$^{(1)}$	87.50/20.83	93.75/54.17	100.00	**65.50**$^{(1)}$	**71.43/74.51**	**93.88/95.10**	98.50
		∅	76.00	80.05/51.43	88.51/74.29	100.00	**84.75**$^{(1)}$	**94.83/14.17**	**97.41/63.33**	100.00	62.14$^{(1)}$	69.00/68.87	94.00/94.34	99.03

Table 17: **Geo**. Property-compliance rate (PCR) and binary property evaluation (BPE) for transitivity on Geo. Each cell is reported from the run with the highest PCR, indexed by (k) when multiple runs exist. PCR and BPE (+/−) are conditioned on successful learning of the injected fact. Port. denotes portability and Eff. injection success. The best value within each group and model is boldfaced and marked by PCR, BPE, or Port.; Eff. < 60 is marked in red. Rows are grouped by injection regime, with the corresponding Baseline shown first.

Prop.	Mode	$\mathcal{K}_p$	Llama3-8B			Llama3-70B			Qwen3-32B		
			Gen. ↑	Spe. ↑	L_{topk} ↑/KL ↓	Gen. ↑	Spe. ↑	L_{topk} ↑/KL ↓	Gen. ↑	Spe. ↑	L_{topk} ↑/KL ↓
Transitivity	baseline		87.73	86.57	93.33/0.57	87.46	86.92	96.21/0.49	87.15	92.40	96.08/0.44
	single_prop		89.12	82.35	91.46/1.02	88.19	85.83	95.67/0.56	88.57	86.91	93.37/1.34
	btsp	$\mathcal{K}_p$	96.18	51.57	73.81/12.08	91.90	85.83	96.39/0.63	89.00	78.08	88.47/5.02
		∅	94.79	50.26	72.97/13.30	93.17	74.60	92.56/4.96	88.62	77.35	88.03/5.31
	pbatch100	$\mathcal{K}_p$	91.08	68.08	83.36/4.57	92.13	85.38	96.42/0.61	88.54	87.82	93.35/1.45
		∅	91.42	67.35	82.91/4.83	92.88	83.90	95.57/0.85	88.14	87.81	93.68/1.46
	baseline, seq B_1		98.09	55.76	77.67/9.45	95.94	61.27	89.85/18.72	93.85	80.46	89.62/4.31
	btsp, seq B_1	$\mathcal{K}_p$	98.83	47.94	69.86/15.49	96.70	55.31	87.33/24.14	94.67	76.20	87.54/5.68
		∅	98.91	46.12	68.73/16.87	96.89	51.15	84.44/41.49	94.17	75.40	87.65/5.63

Table 18: **Geo**. Generalization (Gen.) and specificity (Spe.) for transitivity on Geo. Each cell is reported from the run with the highest PCR, the same run selected in the companion property table, so both describe the same edited model. Gen. and Spe. are measured on B_1, the injected facts. L_{topk}/KL is an auxiliary read on specificity, beyond Spe. itself. Rows are grouped by injection regime, with the corresponding Baseline shown first.