R-JEPA: A RELATIONAL WORLD MODEL THAT PREDICTS BEFORE IT OBSERVES

Anonymous authors
Paper under double-blind review

ABSTRACT

Relational reasoning from a few examples is usually studied by naming the shared relation or scoring candidate answers, while joint embedding predictive architectures (JEPA) hold that predicting a target without negatives is the better objective. These views obscure what a relation does to a state, when that operation can stay fixed, and when predicting the state beats choosing among candidates. We argue that relational reasoning is better understood as an inference cycle in latent space, where a relation induced from example pairs predicts an unobserved state that is reused as the next input and checked against later observations. Based on this view, we propose R-JEPA, a relation reader and a transition operator that returns a state, trained by latent prediction or classification or used without learned parameters. Across knowledge graphs and systems described to language models, the predicted state follows edits of the knowledge, composes over grounded steps and ranks facts missing from a 2019 Wikidata snapshot as well as the true relation does. How consistently roles form directions, measured across seven language models, decides whether the transition must be learned, and latent prediction needs no negatives, while classification with one negative per query trails it by up to 0.13. These results suggest that relational reasoning should be evaluated by prediction before observation, and that latent prediction pays where classification cannot see enough negatives.

1 INTRODUCTION

Told that South Korea goes to Seoul and Japan to Tokyo, a model that knows what the Solar System contains can propose the Sun. The model already knows the Sun. What it has never seen is the correspondence, and no example states it. A few pairs can fit several relations, and the examples themselves select the reading (Section 5.3). Reasoning of this kind is central to human analogy (Gentner, 1983; Holyoak & Thagard, 1995; Mitchell, 2021) and to world models, internal models that predict what has not yet been observed (Craik, 1943; Ha & Schmidhuber, 2018; LeCun, 2022). Recent work has made substantial progress on it through three strategies. Relation embeddings and analogy benchmarks score candidate answers (Mikolov et al., 2013; Ushio et al., 2021a; Yuan et al., 2023), knowledge graph models learn relation operators and compose them over several hops (Bordes et al., 2013; Chen et al., 2019; Hamilton et al., 2018), and task and function vectors steer a language model towards the answer (Hendel et al., 2023; Todd et al., 2024). Joint embedding predictive architectures (JEPA) have in turn made latent prediction without negatives a strong learning signal for world models (LeCun, 2022; Assran et al., 2023; 2025).

Despite this progress, current approaches provide an incomplete account of what a relation does to a state. Scoring and steering return an answer, and knowledge graph operators return states that compose over several hops but are evaluated on held out facts of the graph they were fit to. None tests whether a state predicted for an unnamed relation follows edits of the knowledge, supports the next step and agrees with a later observation where the answer cannot have been learned. Two conditions also remain open, whether a fixed operation suffices or the transition must be learned, and whether predicting the state, as JEPA does, beats choosing among candidates. Both matter for a world model, which must predict before it observes, stay consistent with what it knows and build on its own predictions.

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.

We argue that relational reasoning is better understood as an inference cycle in latent space (Figure 1). A relation is induced from example pairs without a name and applied to the knowledge of a new query to predict the state of its correspondent before it is observed. That state is reused as the next input and checked against later observations. Under this view, state geometry decides whether the transition can stay fixed, and candidate supervision decides whether latent prediction pays. The goal of this paper is to test each operation of the cycle where the answer cannot have been learned, and to identify these two conditions.

Based on this view, we introduce R-JEPA (relational JEPA), a reader that pools example pairs into a relation code, a transition operator that applies the code to the knowledge of a new query, and retrieval that grounds the returned state in a real entity. The operator is trained by latent prediction (the JEPA objective) or classification, or used without learned parameters. We evaluate it on facts deleted before the knowledge is fit, counterfactual edits, systems renamed or invented for the test and facts missing from a 2019 snapshot of Wikidata, with entity embeddings of knowledge graphs and hidden states of language models, mainly Qwen2.5-7B (Yang et al., 2024a) and Pythia-6.9B (Biderman et al., 2023). Three findings follow. First, the predicted state behaves as a world model state. It follows edits of the knowledge (0.94 on invented systems), composes over three steps on a real graph when each state is grounded (0.58, against 0.27 fed back) and ranks facts missing from the snapshot, including facts verified as added later, as well as the true relation does. Second, state geometry decides whether the transition must be learned, as role consistency, the agreement of role directions across domains, predicts the accuracy of a fixed transition (Proposition 1) across seven language models. Third, latent prediction needs no negatives, and classification needs many. With one negative per query, classification trails latent prediction by up to 0.13 on knowledge graphs (Proposition 2), while over the few candidates of language model states it matches or leads.

This paper makes three contributions. First, it reframes relational reasoning as a state transition inferred from examples, whose predicted state can be edited, reused and checked. Second, it introduces R-JEPA and two propositions that predict when the transition needs no learning and what each objective leaves unconstrained. Third, it shows that the premise of JEPA, that predicting a target beats scoring candidates, holds where classification sees too few of many candidates and fails where the candidates are few, with every preregistered criterion reported, held or failed.

2 Related work

Existing work can be grouped into three perspectives on what a relation is and two views of how a predictor should be trained. They have led to substantial progress but leave open whether a relation read from examples predicts a state that can be edited, reused and checked.

Relations as analogies. A first line of work treats a relation as an analogy to be scored, from structure mapping (Gentner, 1983; Falkenhainer et al., 1989) and relations learned before use (Lu et al., 2019; Hill et al., 2019) to vector offsets (Turney, 2006; Mikolov et al., 2013; Levy & Goldberg, 2014), relation embeddings (Ushio et al., 2021b;a) and analogy benchmarks (Czinczoll et al., 2022; Yuan et al., 2023; 2024; Ye et al., 2024). This perspective isolates the correspondence from the facts. However, offsets partly reflect similarity to the query (Linzen, 2016; Rogers et al., 2017), and language models that answer many items (Webb et al., 2023) degrade on counterfactual variants (Lewis & Mitchell, 2024; Wu et al., 2024), so whether a correspondence was inferred or recalled stays open. Our conditions of Section 3 separate the two.

Relations inside language models. A second line locates relations in the forward pass, from in-context learning (Brown et al., 2020; Xie et al., 2022; Garg et al., 2022; von Oswald et al., 2023; Olsson et al., 2022; Min et al., 2022) to task and function vectors (Hendel et al., 2023; Todd et al., 2024) and linear relation decoding (Hernandez et al., 2024; Merullo et al., 2024; Park et al., 2024). They explain how models retrieve familiar facts, and we compare them with R-JEPA on new items.

Relations as operators over knowledge. A third line, the closest to ours, treats a relation as an operator over a knowledge base. Knowledge graph embeddings learn one operator per named relation (Bordes et al., 2013; Sun et al., 2019), query embeddings compose operators over several hops (Hamilton et al., 2018; Ren et al., 2020; Ren & Leskovec, 2020) or ground each hop in candidate entities (Arakelyan et al., 2021), inductive link predictors read a relation from a few support triples

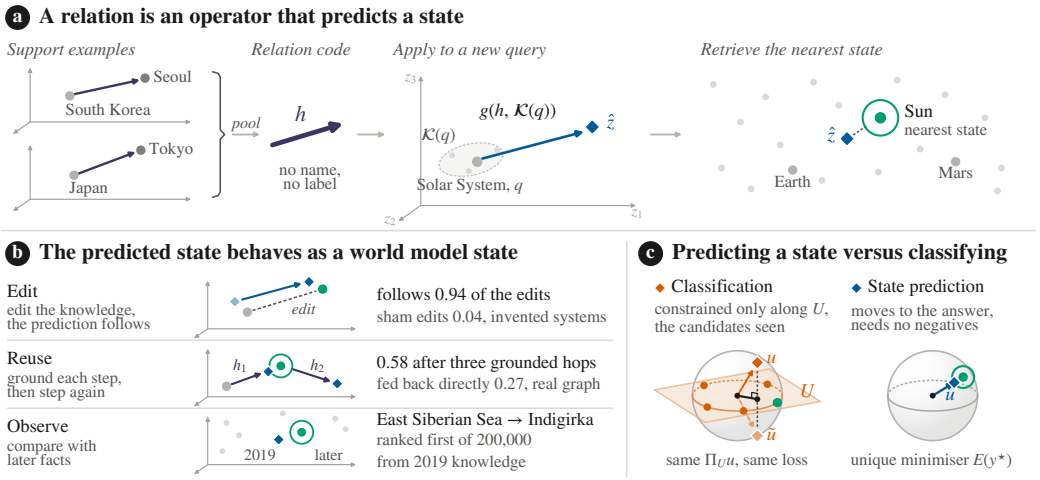

Figure 1: Relational prediction as a state transition. **a** Schematic. A reader pools example pairs into a relation code h that carries no name or label. A transition operator g applies h to what is known about a new query, its knowledge $\mathcal{K}(q)$, and returns a state $\hat{z}$ in the knowledge space, retrieved as the nearest real entity (drawn for the instance without learned parameters, where h is a mean displacement). **b** The predicted state follows edits of the knowledge (sham edits that change no role rarely move it), is grounded and reused as the next input, and is compared with later facts. From example pairs and 2019 knowledge, predicate matching (Section 4) ranks the Indigirka first as a river flowing into the East Siberian Sea, a fact Wikidata recorded only later. **c** Classification constrains the normalised prediction u only along the span U of the candidates it sees (Proposition 2), so u and $\tilde{u}$, which share their projection onto U, receive the same loss. The latent prediction loss is minimised only at the answer's state $E(y^\star)$ and needs no negatives.

(Xiong et al., 2018; Chen et al., 2019; Huang et al., 2022) or transfer to unseen graphs (Teru et al., 2020; Zhu et al., 2021; Lee et al., 2023; Zhou et al., 2023; Galkin et al., 2024), and path and rule methods compose stored relations (Lao & Cohen, 2010; Yang et al., 2017). These methods return reusable states, and the relation vector that MetaR (Chen et al., 2019) reads from support pairs is close to our reader. However, they are evaluated on held out facts of the graph they were fit to, where stored paths may imply the answer, and they do not test whether a prediction follows an edit of the knowledge. In language models, edits of stored facts (Petroni et al., 2019; Geva et al., 2021; 2023; Meng et al., 2022; 2023; Hase et al., 2023) often miss their consequences over several hops (Zhong et al., 2023; Cohen et al., 2024), and models compose, reverse or update stored facts only in part (Press et al., 2023; Yang et al., 2024b; Berglund et al., 2024; Allen-Zhu & Li, 2024; Longpre et al., 2021; Xie et al., 2024). We therefore delete or edit the queried facts before the knowledge is fit, and use facts missing from the 2019 Wikidata snapshot as later observations (Section 5.3), since Wikidata has grown by about half (Wang et al., 2021; Difallah, 2025).

World models, JEPA and negatives. World models predict future states in a latent space (Ha & Schmidhuber, 2018; Hafner et al., 2020; Schrittwieser et al., 2020) and are probed in sequence models (Li et al., 2023; Gurnee & Tegmark, 2024; Vafa et al., 2024). JEPAs predict target representations (LeCun, 2022; Assran et al., 2023; Bardes et al., 2024; Assran et al., 2025; Balestriero & LeCun, 2025), also on graphs, language, and vision and language (Skenderi et al., 2025; Huang et al., 2026; Chen et al., 2026), and handle several targets with a latent variable (LeCun, 2022; Guzman-Rivera et al., 2012; Lee et al., 2016; Rupprecht et al., 2017). Their premise is that contrastive learning (van den Oord et al., 2018; Chen et al., 2020; He et al., 2020) needs negatives that cover the target space, whereas objectives with a stopped target or a regulariser do not (Grill et al., 2020; Chen & He, 2021; Baevski et al., 2022; Bardes et al., 2022; Tian et al., 2021). Negatives also matter through their number and hardness (Arora et al., 2019; Wang & Isola, 2020; Robinson et al., 2021; Ruffinelli et al., 2020). In an open vocabulary, unannotated candidates are often true, so treating them as negatives suppresses correct answers (Elkan & Noto, 2008; Neau, 2026), and regressing onto target embeddings avoids a softmax over a fixed vocabulary (Kumar & Tsvetkov, 2019). On retrieval, a

contrastive loss has been reported to beat latent regression (Chen et al., 2026). We test the premise for relations by varying the negatives from one per query to the full candidate set.

3 RELATIONAL WORLD MODELLING AS PREDICTION BEFORE OBSERVATION

We make each operation of the cycle measurable, testing prediction on correspondences that could not have been learned, dependence on the knowledge by edits and controls, reuse by rollouts, and checking by facts recorded later.

Setting. A model has knowledge $\mathcal{K}$ about entities $\mathcal{E}$, and a state map $E : \mathcal{E} \to \mathbb{R}^d$ derived from that knowledge and frozen before any relation is learned. In a knowledge graph, $\mathcal{K}$ is a set of facts (x, p, y) and E is an entity table fit to them. In a language model, $\mathcal{K}$ is what the model has read, possibly together with a description given in context, and $E(x)$ is the hidden state of x's name in that context. An episode gives k supports $S = \{(a_i, b_i)\}_{i=1}^{k}$ that instantiate one unnamed relation R, and a query q. In analogy data the query is a system such as the Solar System, the candidates are its components, and a role is the part a component plays in its system, such as the centre. The model outputs a state $\hat{z}$, and its answer is the nearest candidate,

$$\hat{y} = \arg\max_{c \in \mathcal{C}(q)} \cos\big(\hat{z}, E(c)\big), \tag{1}$$

where $\mathcal{C}(q)$ is either every entity of the graph or the known components of the query system. The gold set $Y^\star(q)$ contains every candidate $y^\star$ that stands in R to q. The relation is never named or labelled, and it is never seen with the query's domain during training.

Unobserved correspondences. An episode tests inference only if the correspondence $(q, y^\star)$ could not have been learned. We use four conditions (Appendix A). Under *deletion*, every fact that links q and $y^\star$ is removed before the knowledge is fit, so the states never encoded the correspondence and no neighbour lookup returns it. Under a *counterfactual edit*, the knowledge is changed so that another entity y' holds the role. The correct prediction is then y', which no training signal associated with q. In a *renamed system*, a query system of SCAR, a benchmark of analogies between scientific systems (Yuan et al., 2023), and its components are renamed with pronounceable nonce words throughout a textual description. The only knowledge about the names is that description, although the kind of system (a business, a cell) remains familiar. In an *invented system*, the system, its kind and its nine components are invented for the test, and their roles are stated only relationally in ordinary English. Query descriptions use sentence templates and kind nouns different from those of the supports and of training. Nonce words are the classical way to separate a rule from stored items (Berko, 1958), and new items remove the contamination problem of public benchmarks (Sainz et al., 2023). These conditions change the knowledge the model is given and keep the task format (Wu et al., 2024; Lewis & Mitchell, 2024). Facts that a knowledge snapshot lacks and that were recorded afterwards give a natural fifth condition, a time split, in which the later observation is real (Section 5.3). Throughout, whether pairs share a relation is decided by the data and never by us, through a stored property or analogous properties listed by AnalogyKB on graphs (Yuan et al., 2024), the human mappings of SCAR, and the construction of the generated systems.

Controls for knowledge dependence. With the description removed (*names only*), accuracy must fall to chance. After an edit that gives the gold's description to a distractor, the prediction must *follow* it. A *sham* edit that changes no statement about roles must rarely change it, reordering the description must never move it to the distractor, and shuffling the supports must remove the ability. On *role swap* pairs, one query gets supports of two roles with disjoint gold sets, so a predictor that ignores the supports gets at most one right, and we report how often both are right.

Reuse. A predicted state can be fed back as the query of the next step (*latent* reuse), or first replaced by the entity it retrieves (*grounded* reuse). Feeding the true intermediate entity (teacher forcing) bounds both, and the accuracy that latent reuse loses against it is the *drift*.

Metrics. We report top-1 accuracy among the candidates, with the chance level, and on the time split the mean reciprocal rank (MRR). Ties with a wrong candidate count as errors. On graphs, other gold answers are filtered. Learned models are trained only on episodes from training domains,

topics or fields and use three seeds. Differences are given with paired bootstrap intervals over query systems, query entities or chains.

4 R-JEPA

Reader, operator, retrieval. R-JEPA has three parts, one for each operation of the cycle. A reader r_ϕ, a permutation invariant set function (Zaheer et al., 2017), pools the support pairs into a relation code $h = r_\phi(\{(E(a_i), E(b_i))\}_i)$, so the relation is induced from examples and never looked up by name. A transition operator g_θ applies h to the knowledge of the query and returns a state $\hat{z} = g_\theta(h, \mathcal{K}(q))$ in the space of E. Retrieval by equation 1 compares $\hat{z}$ with real entities and grounds it in one. This architecture alone lets $\hat{z}$ be reused, change with $\mathcal{K}$ and be compared with a later observation, even without learned parameters. R-JEPA names this architecture, whose operator predicts a state in the space of E as the predictor of a JEPA does. The objective decides only how g_θ is fit, so comparing objectives on one architecture isolates the effect of latent prediction. Rollouts ground each step in the retrieved entity, as a latent world model encodes each new observation. Where the knowledge is a graph, a knowledge model fit to it (Appendix B) has a decoder $f(y \mid x, p)$ that scores every entity y as the object of subject x and predicate p, and its expected state $m_p(x) = \sum_y f(y \mid x, p)\, E(y)$ is the transition along p.

Instances. Two operators have no learned parameters. The **centred offset** uses the role vector $h = \frac{1}{k}\sum_i \big(E(b_i) - \bar{m}_{a_i}\big)$, where $\bar{m}_a$ is the mean state of system a's components, predicts $\hat{z} = \bar{m}_q + h$ and scores candidates by $\cos(E(c) - \bar{m}_q, h)$. **Predicate matching** selects the stored predicate $\hat{p} = \arg\max_p \frac{1}{k}\sum_i \log f(b_i \mid a_i, p)$ that best explains the supports and predicts $\hat{z} = m_{\hat{p}}(q)$. Learned operators attend over the states of the query's components or graph neighbours (**component attention**, and **neighbour attention** with the linking edge) (Vinyals et al., 2015), or over the transitions $m_p(q)$ (**transition attention** and its variants), so that the operator is a transition model of the knowledge. A learned centred offset adds a projection of the states and a linear map of h. Appendices B and D.1 give the details and each result's operator. Classical references are the prototype $\frac{1}{k}\sum_i E(b_i)$ (Snell et al., 2017) and the offset $E(q) + \frac{1}{k}\sum_i \big(E(b_i) - E(a_i)\big)$ (Mikolov et al., 2013).

Objectives. Learned readers and operators are trained on identical batches with one of two losses,

$$\mathcal{L}_{\text{Pred}} = 1 - \cos\big(\hat{z}, \operatorname{sg} E(y^\star)\big), \qquad \mathcal{L}_{\text{Cls}} = -\log \frac{\sum_{y \in Y^\star} e^{s_y/\tau}}{\sum_{c \in \mathcal{C}(q)} e^{s_c/\tau}}, \tag{2}$$

with $s_c = \cos(\hat{z}, E(c))$, $\tau = 0.1$ and sg the stop gradient. Pred is the JEPA objective. It regresses the predicted state onto a stopped target, the frozen state table or, when the operator also learns a projection of the states, an exponential moving average (EMA) of that projection (Grill et al., 2020). Cls is the classification control, a softmax cross entropy over the candidates with every gold as a positive, and uses the same scores as retrieval. On real data, half of each batch consists of role swap pairs (Section 3), so ignoring the supports is penalised.

Two further variants follow the JEPA design where Pred departs from it. **Pred+z** adds a discrete latent variable (LeCun, 2022). The operator has four output heads that share all other parameters, the head closest to the target takes weight 0.95 of $\mathcal{L}_{\text{Pred}}$ and the others share the rest (Guzman-Rivera et al., 2012; Rupprecht et al., 2017), and retrieval scores each candidate by its best head. For scarce negatives, $\textbf{Cls}_n$ ranks the golds against n candidates sampled per step, and $\textbf{Cls}_\text{b}$ against the other items' targets in the batch only, as when no candidate set is known. These are the InfoNCE loss with sampled and with in-batch negatives (van den Oord et al., 2018; Chen et al., 2020).

What the states and the objectives must provide. Two propositions organise the experiments, and Appendix C gives the proofs.

Proposition 1 (retrieval without learning). For the query system, let each centred candidate state $E(c) - \bar{m}_q$ be a positive multiple of $r_{\rho(c)} + \epsilon_c$, where $\rho(c)$ is the role of c. For each support, let $E(b_i) - \bar{m}_{a_i}$ be a positive multiple of $r_\rho + \epsilon_i$, where ρ is the role the supports share. Suppose every role direction r_ρ is a unit vector, $\langle r_\rho, r_{\rho'} \rangle \le 1 - \gamma$ for $\rho \neq \rho'$, and every noise term has norm at most δ. If $\gamma > 6\delta$, the centred offset retrieves a holder of ρ.

A fixed operation therefore suffices when roles are consistent directions with a margin, which is the geometry that analogies by vector offset rely on (Mikolov et al., 2013; Park et al., 2024). When the states lack it, the operator has to be learned, for example when domains express a role through different predicates or when a layer does not separate roles. The objective that trains it then matters, because each objective leaves different directions free.

Proposition 2 (what classification leaves free). Let the candidate states $E(c)$ be unit vectors, let $U = \mathrm{span}\{E(c) : c \in \mathcal{C}(q)\}$ with $\dim U = n_U < d - 1$, and let $u = \hat{z}/\|\hat{z}\|$. Then $\mathcal{L}_{\mathrm{Cls}}$ depends on u only through the projection $\Pi_U u$. When $\|\Pi_U u\| < 1$, every unit vector with the same projection attains the same loss, and these vectors form a sphere of dimension $d - n_U - 1$ parallel to $U^{\perp}$. In contrast, $\mathcal{L}_{\mathrm{Pred}}$ has the unique minimiser $u = E(y^{\star})$, and for any direction w read at the next step, $|\langle u - E(y^{\star}), w\rangle| \le \sqrt{2\mathcal{L}_{\mathrm{Pred}}}\,\|w\|$.

When $\hat{z}$ is reused as the next query, the next step reads directions outside U. Classification does not train these components of u, and latent prediction bounds them. The same holds when classification sees only a sampled subset $\mathcal{S} \subset \mathcal{C}(q)$. It then constrains u only within the span of $\mathcal{S}$, and directions that separate the gold from unsampled candidates stay free. Latent prediction needs no candidates, so it can gain when negatives are few relative to the answer space. An operator whose output is restricted to convex combinations of known states (transition attention) removes much of the free component by construction. Section 5.4 tests what trained models do.

5 Experiments

5.1 Setup

Knowledge and states. We use three kinds of knowledge (Appendix A). The *controlled world* has twelve domains of 30 systems whose roles are unnamed bundles of properties. A share α of role facts uses predicate words shared across domains, and four domains are held out. The *Wikidata graphs* are WikiTopics (Zhou et al., 2023), topic subgraphs of Wikidata5M (Wang et al., 2021) with two splits, A and B, that hold out different topics and take roles from AnalogyKB (Yuan et al., 2024), and MQuAKE, whose questions over several hops have counterfactually edited facts (Zhong et al., 2023). Graph states are the entity table of a knowledge model fit to the graph, possibly after deletion, and then frozen. For *language models*, SCAR roles are the connected components of SCAR's human mappings between systems (Yuan et al., 2023), renamed systems are 117 described SCAR query systems, and invented systems are 200 test systems (360 episodes, 9 candidates) and 400 training systems. A component's state is the mean hidden state of its name in "In {system}, the {component}", preceded by the description for renamed and invented systems. We use Qwen layer 21 and Pythia layer 8, fixed by a preregistered rule.

Baselines. Baselines are the prototype, the offset, the neighbour mean (mean of the query's graph neighbours), personalised PageRank (Haveliwala, 2002) and a decoder given the true relation, and Appendix D.3 compares the models' own mechanisms (Hendel et al., 2023; Todd et al., 2024).

5.2 Prediction before observation

With an explicit decoder, identifying the relation is retrieval. When the supports instantiate a stored predicate (Table 1, top), predicate matching finds it and the decoder applies it. With the queried facts deleted before the knowledge is fit, it reaches the accuracy of the decoder given the true relation (0.29 top-1 among 10 to 20 thousand entities), so the ceiling is the knowledge's own, and learned operators stay below it. When supports and query realise the role through different predicates, no stored predicate fits both, and all operators trail graph proximity (boundary row).

In language model states, relations transfer without learning. The centred offset finds the component that plays the supports' role in a system from another field (0.57 Qwen, 0.51 Pythia, chance 0.06) and in systems invented for the test (0.88 and 0.41, chance 0.11, and 0.10 and 0.11 once the description is removed). It is ahead of the models' own mechanisms except on invented systems, where the model answering in context reads the full descriptions (Appendix D.3), and trained operators help mainly on invented systems and on Pythia's renamed systems.

Table 1: **Prediction before observation.** Top-1 on relations, domains or systems held out from training (the three-hop rows compose stored facts), grouped by how knowledge is represented. Chance is the share of golds among the candidates. Learned operators use three seeds. Shading marks the best two per row. Baselines are the ^{p}prototype, ^{o}offset, ^{n}neighbour mean and ^{r}personalised PageRank, and R-JEPA without learning is the centred offset or †predicate matching (Appendix D.1).

		No learning		Learned R-JEPA	
Setting	Chance	Baseline	R-JEPA	Cls	Pred
Explicit knowledge with a decoder					
Deleted facts, split A	<.001	0.18^{p}	0.29†	0.21	0.20
Deleted facts, split B	<.001	0.20^{o}	0.26†	0.19	0.18
Language model states (Qwen layer 21, Pythia layer 8)					
Roles across fields, Qwen	0.06	0.52^{p}	0.57	0.52	0.44
Roles across fields, Pythia	0.06	0.49^{p}	0.51	0.41	0.38
Renamed systems, Qwen	0.07	0.31^{p}	0.34	0.31	0.28
Renamed systems, Pythia	0.07	0.14^{p}	0.08	0.23	0.20
Invented systems, Qwen	0.11	0.87^{p}	0.88	0.99	0.98
Invented systems, Pythia	0.11	0.41^{p}	0.41	0.74	0.66
Entity tables without a decoder, and Qwen layer 7					
Controlled world, α=1	0.19	0.67^{p}	0.83	0.95	0.97
Controlled world, α=0	0.19	0.37^{p}	0.50	0.72	0.76
Graph states, no decoder, A	<.001	0.38^{n}	0.32	0.72	0.69
Graph states, no decoder, B	<.001	0.26^{n}	0.24	0.43	0.51
Renamed systems, Qwen layer 7	0.07	0.09^{p}	0.13	0.13	0.13
Reusing the predicted state as the next query					
Two steps, controlled world	0.007	0.20^{o}	0.28	0.35	0.78
Two steps, invented, Qwen	0.11	0.91^{p}	0.71	1.00	0.95
Two steps, invented, Pythia	0.11	0.30^{p}	0.19	0.59	0.48
Three hops, real graph, latent	0.02	0.04^{o}	0.48†	0.26	0.27
Three hops, real graph, grounded	0.02	0.05^{o}	0.69†	0.37	0.58
Boundary					
Roles across predicates, deleted, A	<.001	0.22^{r}	0.01†	0.03	0.03

In entity tables without a decoder, the operator must be learned. There the centred offset falls short, and learned operators recover transfer (controlled world 0.83 and 0.50 without learning, 0.95 to 0.97 and 0.72 to 0.76 learned), with neither objective ahead in every setting. In an early Qwen layer (7) the centred offset drops from 0.34 to 0.13, and learning does not recover it.

5.3 The predicted state follows the knowledge, is reused and is checked later

Edits. The same description yields different answers under different supports (Figure 2a), and edits move the prediction by content (Figure 2c). Giving the gold's description to a distractor moves 0.94 of correct predictions on invented systems and 0.65 on renamed systems, while sham edits change 0.04 and 0.09. Reordering sentences never moves a prediction to the distractor (0 of 40 under each of three orders). On MQuAKE, the learned rollout follows a counterfactual edit of the graph in 0.78 of chains (three seeds) and leaves 0.97 of unrelated chains unchanged, and a sham that relearns the unchanged facts changes 0.03 of the chains that were correct before (0.17 of all chains, above our preregistered 0.10). Under familiar names, in-context edits move no prediction, and rank one model editing (ROME, Meng et al., 2022) moves 0.03 while its sham changes 0.17, as stored knowledge competes with context (Longpre et al., 2021; Xie et al., 2024), so we claim knowledge following only for graphs and described systems.

Reuse. Proposition 2 predicts that latent prediction bounds what the next step reads. This holds in the controlled world (two latent steps, Pred 0.78, Cls 0.35, centred offset 0.28, chance 0.007). In invented systems classification is more accurate after two steps (Qwen 1.00 against 0.95, Pythia 0.59 against 0.48), but a prototype that ignores the first step already reaches 0.91 and 0.30, so this task

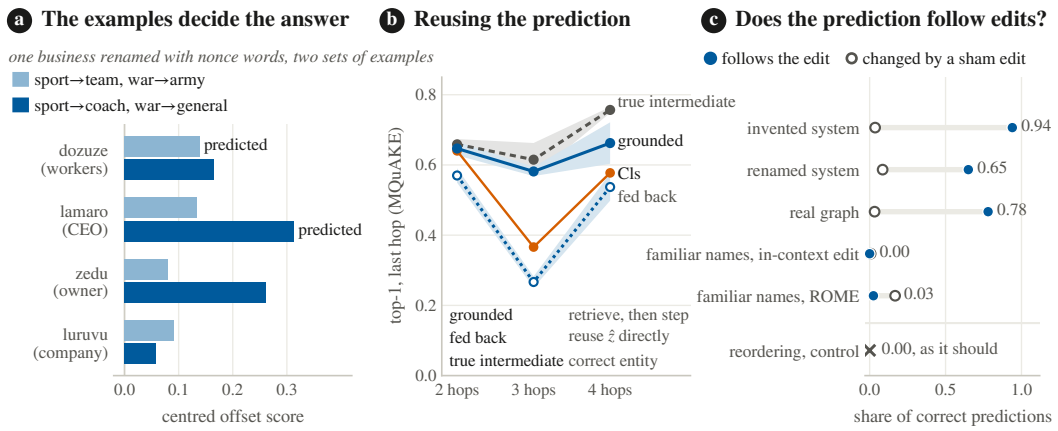

Figure 2: **a** Centred offset scores of one renamed system's components under two support sets (Qwen layer 21), original names in parentheses. **b** Composition on MQuAKE for Pred and grounded Cls (three seeds, band s.d.), where 2, 3 and 4 hops are different questions. **c** Correct predictions that follow an edit (filled, share given), change under a sham edit (open, on the graph among chains correct before) and move to the distractor under reordering (crosses).

tests chaining weakly. On MQuAKE, three latent steps keep 0.27 of chains with latent prediction and 0.26 with classification (Figure 2b). The frozen decoder amplifies the latent state's small error. Grounding each step in the retrieved entity removes the drift, with 0.58 after three hops against 0.62 from the true intermediate entity (classification 0.37). Training the operator on its own rollouts lowers the drift only from 0.35 to 0.27. Predicate matching composes better without learning (0.48 latent, 0.69 grounded), because chain facts are stored and naming the relation recalls them.

Later observations. With knowledge fit to a 2019 Wikidata snapshot of 200,000 entities, R-JEPA reads each relation from five pairs and ranks facts the snapshot lacks (Appendix D.6). Predicate matching equals the decoder given the true relation on the 18,835 of them whose query the snapshot does not answer (MRR 0.418, the same rank for 99.6% of them), on 825 checked to have been added after 2019 (0.317), and for subjects of a type the relation was never recorded for (0.345, learned operators 0.21 to 0.25).

5.4 When the operator must be learned, and when the JEPA objective helps

Role geometry. For every support system, let v be the normalised difference between the role holder's state and the system's mean state. Role consistency κ is the mean cosine between the v of the same role in different domains minus that between different roles. Within each family of states, κ orders the centred offset's accuracy (Figure 3a). Removing shared predicates lowers κ from 0.31 to 0.16 and accuracy from 0.83 to 0.50 in the controlled world, where learned operators reach 0.76 and 0.97. Across seven language models at seven depths each, κ of the SCAR states predicts accuracy, as preregistered (Spearman 0.45 over 49 points, 95% interval [0.17, 0.67] with models resampled), and the final layer of most models falls off the trend. Values of κ are comparable only within a family of states (Appendix D.3).

Scarce negatives. Proposition 2 implies that classification constrains the prediction only along the candidates it sees. We withheld candidates from classification while keeping the evaluation unchanged (Figure 3b). With about 13,000 candidates per query on the topic graphs, classification with one negative falls 0.12 to 0.13 below latent prediction and trails it in every seed. With 16 negatives or all candidates it is within seed variation of latent prediction on split A and still trails it on split B. With small or typed answer spaces one negative suffices (MQuAKE, 177 candidates, and invented systems, 9 components). On the 200,000 entity graph of the time split, the two objectives perform alike with the full candidate set (MRR 0.370 and 0.369 for queries the snapshot does not answer), as Proposition 2 expects. With 93 relations held out, 16 to 256 negatives lead latent prediction by 0.03

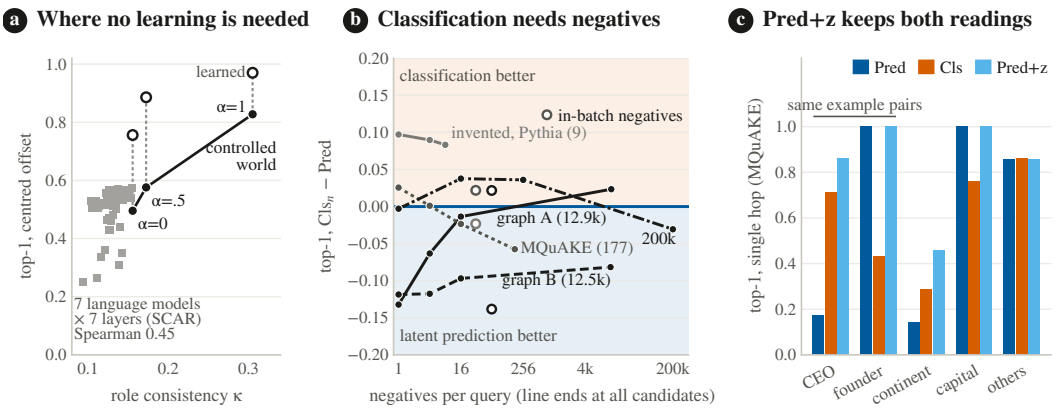

Figure 3: **a** Role consistency κ and the centred offset's top-1 (Proposition 1). Dotted lines rise to the best learned operator (open circles). **b** Top-1 of classification with n negatives per query minus latent prediction (three seeds). Lines end at the full candidate set, open markers use in-batch negatives, and 200k marks held out relations of the time split. **c** Relations whose supports coincide, as founders are often CEOs. Pred commits to one reading, Cls blurs them, Pred+z keeps both.

to 0.06 on held out and seen relations alike (Appendix D.5). One negative per query is thus too few on large graphs, and how many suffice depends on the graph.

A latent variable for ambiguous relations. On graph states a latent variable scores 0.09 to 0.10 below Pred in top-1, as a head placed on another answer can win, so we examined where it helps (exploratory). On MQuAKE, where candidates are typed by relation, single step accuracy rises from 0.68 (Pred) and 0.62 (Cls) to 0.72, mostly where the supports leave the relation ambiguous (Figure 3c). For founder and CEO pairs, which often name the same person, Pred commits to one reading, Cls blurs them, and the latent variable keeps both (1.00 and 0.86).

Where the JEPA objective does not help. With enumerated candidates, classification matches or exceeds latent prediction in every language model setting, as reported for retrieval (Chen et al., 2026), and entity states trained with it neither fit the facts nor build role geometry (Appendix D.5).

6 Discussion and Conclusion

Our results support treating relational reasoning as an inference cycle over states, in which a state predicted from examples for an unobserved correspondence moves with the knowledge, supports the next step once grounded and agrees with facts recorded later. Grounding plays the role that each new observation plays in a latent world model (Hafner et al., 2020). On a real graph, feeding the state back loses 0.35 of accuracy over three hops, training on the model's own rollouts still loses 0.27, and grounding each step loses 0.03.

Limitations. A fixed transition fails when one role is realised by different predicates, familiar names resist edits, latent prediction leads mainly when classification sees too few of many candidates, and the sham control of graph edits passes only on chains correct before the edit. Our states come from frozen models, the entity tables of one knowledge model per graph and language models of 1 to 8 billion parameters, and they are never trained together with the operator. Descriptions are in English, the analogies are SCAR's 235 episodes and the systems we generated, and 1,292 facts of the time split are verified as added later. Whether the two conditions hold for larger models and for states trained with the operator remains open.

Relational models should therefore read relations into operators over states, ground their rollouts and train by latent prediction where candidates cannot be enumerated. Relational reasoning should be evaluated where the answer cannot have been learned. Predicting what has not yet been observed is a concrete test of relational understanding.

AI USE STATEMENT

In this work, we used generative AI tools for writing and running experiment code, for generating the figures of the paper, and for polishing its text. We have reviewed all AI-assisted work, and every reported number is read from result files by scripts included in the supplementary material. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses public data, namely graphs derived from Wikidata, current Wikidata statements (CC0), SCAR and MQuAKE. It uses open-weight language models, Qwen2.5-7B and Pythia-6.9B throughout and five more for role consistency (Section 5.4). It involves no human subjects and no personal data beyond what public encyclopedic entries contain. Counterfactual edits are applied only to local copies of knowledge graphs and model weights. We release no edited model. Knowledge editing methods can be misused to insert false facts. In the settings we tested, such edits rarely propagated to relational predictions about familiar entities, but that finding should not be read as a safety guarantee.

REPRODUCIBILITY STATEMENT

Section 3 and Appendix A specify every evaluation condition. Appendix A gives the rules used to build each derived dataset. Each dataset ships with a manifest of source and output hashes and an audit of every filter. Appendix B gives the architectures, training settings and baselines, and Appendix C proves both propositions. Appendix E lists every preregistered criterion and its outcome, including failures. The anonymous supplementary material contains the code, the builders and manifests of the derived datasets, the smaller derived datasets, and the result files and scripts that regenerate every table, figure and number.

REFERENCES

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.1, knowledge storage and extraction. In *Proceedings of ICML*, pp. 1067–1077, 2024.

Erik Arakelyan, Daniel Daza, Pasquale Minervini, and Michael Cochez. Complex query answering with neural link predictors. In *Proceedings of ICLR*, 2021.

Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. A theoretical analysis of contrastive unsupervised representation learning. In *Proceedings of ICML*, pp. 5628–5637, 2019.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of CVPR*, pp. 15619–15629, 2023.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, et al. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. data2vec: A general framework for self-supervised learning in speech, vision and language. In *Proceedings of ICML*, pp. 1298–1312, 2022.

Randall Balestriero and Yann LeCun. LeJEPA: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025.

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *Proceedings of ICLR*, 2022.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Transactions on Machine Learning Research*, 2024.

Lukas Berglund, Meg Tong, Maximilian 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 *Proceedings of ICLR*, 2024.

Jean Berko. The child's learning of English morphology. *Word*, 14(2–3):150–177, 1958.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, et al. Pythia: A suite for analyzing large language models across training and scaling. In *Proceedings of ICML*, pp. 2397–2430, 2023.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in NeurIPS*, volume 26, 2013.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, et al. Language models are few-shot learners. In *Advances in NeurIPS*, volume 33, pp. 1877–1901, 2020.

Delong Chen, Mustafa Shukor, Théo Moutakanni, Willy Chung, et al. VL-JEPA: Joint embedding predictive architecture for vision-language. In *Proceedings of ICLR*, 2026.

Mingyang Chen, Wen Zhang, Wei Zhang, Qiang Chen, and Huajun Chen. Meta relational learning for few-shot link prediction in knowledge graphs. In *Proceedings of EMNLP*, pp. 4217–4226, 2019.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of ICML*, pp. 1597–1607, 2020.

Xinlei Chen and Kaiming He. Exploring simple Siamese representation learning. In *Proceedings of CVPR*, pp. 15750–15758, 2021.

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.

Kenneth J. W. Craik. *The Nature of Explanation*. Cambridge University Press, Cambridge, 1943.

Tamara Czinczoll, Helen Yannakoudakis, Pushkar Mishra, and Ekaterina Shutova. Scientific and creative analogies in pretrained language models. In *Findings of EMNLP*, pp. 2094–2100, 2022.

Djellel Difallah. WikiRAG: Revisiting Wikidata KGC datasets with community updates and retrieval-augmented generation. In *Proceedings of KDD*, pp. 5391–5401, 2025.

Charles Elkan and Keith Noto. Learning classifiers from only positive and unlabeled data. In *Proceedings of KDD*, pp. 213–220, 2008.

Brian Falkenhainer, Kenneth D. Forbus, and Dedre Gentner. The structure-mapping engine: Algorithm and examples. *Artificial Intelligence*, 41(1):1–63, 1989.

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

Shivam Garg, Dimitris Tsipras, Percy S. Liang, and Gregory Valiant. What can transformers learn in-context? A case study of simple function classes. In *Advances in NeurIPS*, volume 35, pp. 30583–30598, 2022.

Dedre Gentner. Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7(2): 155–170, 1983.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of EMNLP*, pp. 5484–5495, 2021.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. Dissecting recall of factual associations in auto-regressive language models. In *Proceedings of EMNLP*, pp. 12216–12235, 2023.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, et al. Bootstrap your own latent - a new approach to self-supervised learning. In *Advances in NeurIPS*, volume 33, pp. 21271–21284, 2020.

Wes Gurnee and Max Tegmark. Language models represent space and time. In *Proceedings of ICLR*, 2024.

Abner Guzman-Rivera, Dhruv Batra, and Pushmeet Kohli. Multiple choice learning: Learning to produce multiple structured outputs. In *Advances in NeurIPS*, volume 25, 2012.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in NeurIPS*, volume 31, 2018.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *Proceedings of ICLR*, 2020.

Will Hamilton, Payal Bajaj, Marinka Zitnik, Dan Jurafsky, and Jure Leskovec. Embedding logical queries on knowledge graphs. In *Advances in NeurIPS*, volume 31, 2018.

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 NeurIPS*, volume 36, pp. 17643–17668, 2023.

Taher H. Haveliwala. Topic-sensitive PageRank. In *Proceedings of WWW*, pp. 517–526, 2002.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of CVPR*, pp. 9729–9738, 2020.

Roee Hendel, Mor Geva, and Amir Globerson. In-context learning creates task vectors. In *Findings of EMNLP*, pp. 9318–9333, 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 *Proceedings of ICLR*, 2024.

Felix Hill, Adam Santoro, David G. T. Barrett, Ari S. Morcos, and Timothy Lillicrap. Learning to make analogies by contrasting abstract relational structure. In *Proceedings of ICLR*, 2019.

Keith J. Holyoak and Paul Thagard. *Mental Leaps: Analogy in Creative Thought*. MIT Press, Cambridge, MA, 1995.

Hai Huang, Yann LeCun, and Randall Balestriero. LLM-JEPA: Large language models meet joint embedding predictive architectures. In *Proceedings of ICLR*, 2026.

Qian Huang, Hongyu Ren, and Jure Leskovec. Few-shot relational reasoning via connection subgraph pretraining. In *Advances in NeurIPS*, volume 35, pp. 6397–6409, 2022.

Sachin Kumar and Yulia Tsvetkov. Von Mises-Fisher loss for training sequence to sequence models with continuous outputs. In *Proceedings of ICLR*, 2019.

Ni Lao and William W. Cohen. Relational retrieval using a combination of path-constrained random walks. *Machine Learning*, 81(1):53–67, 2010.

Yann LeCun. A path towards autonomous machine intelligence. OpenReview preprint, `https://openreview.net/forum?id=BZ5a1r-kVsf`, 2022.

Jaejun Lee, Chanyoung Chung, and Joyce Jiyoung Whang. InGram: Inductive knowledge graph embedding via relation graphs. In *Proceedings of ICML*, pp. 18796–18809, 2023.

Stefan Lee, Senthil Purushwalkam, Michael Cogswell, Viresh Ranjan, David Crandall, and Dhruv Batra. Stochastic multiple choice learning for training diverse deep ensembles. In *Advances in NeurIPS*, volume 29, 2016.

Omer Levy and Yoav Goldberg. Linguistic regularities in sparse and explicit word representations. In *Proceedings of CoNLL*, pp. 171–180, 2014.

Martha Lewis and Melanie Mitchell. Using counterfactual tasks to evaluate the generality of analogical reasoning in large language models. In *Proceedings of CogSci*, volume 46, 2024.

Kenneth Li, Aspen K. Hopkins, David Bau, Fernanda B. Viégas, Hanspeter Pfister, and Martin Wattenberg. Emergent world representations: Exploring a sequence model trained on a synthetic task. In *Proceedings of ICLR*, 2023.

Tal Linzen. Issues in evaluating semantic spaces using word analogies. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pp. 13–18, 2016.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. Entity-based knowledge conflicts in question answering. In *Proceedings of EMNLP*, pp. 7052–7063, 2021.

Hongjing Lu, Ying Nian Wu, and Keith J. Holyoak. Emergence of analogy from relation learning. *Proceedings of the National Academy of Sciences*, 116(10):4176–4181, 2019.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in NeurIPS*, 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 *Proceedings of ICLR*, 2023.

Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. Language models implement simple Word2Vec-style vector arithmetic. In *Proceedings of NAACL*, pp. 5030–5047, 2024.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of NAACL*, pp. 746–751, 2013.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of EMNLP*, pp. 11048–11064, 2022.

Melanie Mitchell. Abstraction and analogy-making in artificial intelligence. *Annals of the New York Academy of Sciences*, 1505(1):79–101, 2021.

Maëlic Neau. RelateAnything: Real-time open-vocabulary relation prediction from any inputs. *arXiv preprint arXiv:2609.12552*, 2026.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *Proceedings of ICML*, pp. 39643–39666, 2024.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language models as knowledge bases? In *Proceedings of EMNLP*, pp. 2463–2473, 2019.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In *Findings of EMNLP*, pp. 5687–5711, 2023.

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

Hongyu Ren, Weihua Hu, and Jure Leskovec. Query2box: Reasoning over knowledge graphs in vector space using box embeddings. In *Proceedings of ICLR*, 2020.

Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. In *Proceedings of ICLR*, 2021.

Anna Rogers, Aleksandr Drozd, and Bofang Li. The (too many) problems of analogical reasoning with word vectors. In *Proceedings of *SEM*, pp. 135–148, 2017.

Daniel Ruffinelli, Samuel Broscheit, and Rainer Gemulla. You CAN teach an old dog new tricks! on training knowledge graph embeddings. In *Proceedings of ICLR*, 2020.

Christian Rupprecht, Iro Laina, Robert DiPietro, Maximilian Baust, Federico Tombari, Nassir Navab, and Gregory D. Hager. Learning in an uncertain world: Representing ambiguity through multiple hypotheses. In *Proceedings of ICCV*, pp. 3591–3600, 2017.

Oscar Sainz, Jon Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. NLP evaluation in trouble: On the need to measure LLM data contamination for each benchmark. In *Findings of EMNLP*, pp. 10776–10787, 2023.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, et al. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

Geri Skenderi, Hang Li, Jiliang Tang, and Marco Cristani. Graph-level representation learning with joint-embedding predictive architectures. *Transactions on Machine Learning Research*, 2025.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in NeurIPS*, volume 30, 2017.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. RotatE: Knowledge graph embedding by relational rotation in complex space. In *Proceedings of ICLR*, 2019.

Komal Teru, Etienne Denis, and Will Hamilton. Inductive relation prediction by subgraph reasoning. In *Proceedings of ICML*, pp. 9448–9457, 2020.

Yuandong Tian, Xinlei Chen, and Surya Ganguli. Understanding self-supervised learning dynamics without contrastive pairs. In *Proceedings of ICML*, pp. 10268–10278, 2021.

Eric Todd, Millicent L. Li, Arnab Sen Sharma, Aaron Mueller, Byron C. Wallace, and David Bau. Function vectors in large language models. In *Proceedings of ICLR*, 2024.

Peter D. Turney. Similarity of semantic relations. *Computational Linguistics*, 32(3):379–416, 2006.

Asahi Ushio, Jose Camacho-Collados, and Steven Schockaert. Distilling relation embeddings from pretrained language models. In *Proceedings of EMNLP*, pp. 9044–9062, 2021a.

Asahi Ushio, Luis Espinosa Anke, Steven Schockaert, and Jose Camacho-Collados. BERT is to NLP what AlexNet is to CV: Can pre-trained language models identify analogies? In *Proceedings of ACL*, pp. 3609–3624, 2021b.

Keyon Vafa, Justin Y. Chen, Ashesh Rambachan, Jon Kleinberg, and Sendhil Mullainathan. Evaluating the world model implicit in a generative model. In *Advances in NeurIPS*, volume 37, pp. 26941–26975, 2024.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In *Advances in NeurIPS*, volume 28, 2015.

Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *Proceedings of ICML*, pp. 35151–35174, 2023.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *Proceedings of ICML*, pp. 9929–9939, 2020.

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. *Transactions of the Association for Computational Linguistics*, 9:176–194, 2021.

Taylor Webb, Keith J. Holyoak, and Hongjing Lu. Emergent analogical reasoning in large language models. *Nature Human Behaviour*, 7(9):1526–1541, 2023.

Zhaofeng Wu, Linlu Qiu, Alexis Ross, Ekin Akyürek, et al. Reasoning or reciting? Exploring the capabilities and limitations of language models through counterfactual tasks. In *Proceedings of NAACL*, pp. 1819–1862, 2024.

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. In *Proceedings of ICLR*, 2024.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit Bayesian inference. In *Proceedings of ICLR*, 2022.

Wenhan Xiong, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. One-shot relational learning for knowledge graphs. In *Proceedings of EMNLP*, pp. 1980–1990, 2018.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024a.

Fan Yang, Zhilin Yang, and William W. Cohen. Differentiable learning of logical rules for knowledge base reasoning. In *Advances in NeurIPS*, volume 30, 2017.

Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. Do large language models latently perform multi-hop reasoning? In *Proceedings of ACL*, pp. 10210–10229, 2024b.

Xiao Ye, Andrew Wang, Jacob Choi, Yining Lu, et al. AnaloBench: Benchmarking the identification of abstract and long-context analogies. In *Proceedings of EMNLP*, pp. 13060–13082, 2024.

Siyu Yuan, Jiangjie Chen, Xuyang Ge, Yanghua Xiao, and Deqing Yang. Beneath surface similarity: Large language models make reasonable scientific analogies after structure abduction. In *Findings of EMNLP*, pp. 2446–2460, 2023.

Siyu Yuan, Jiangjie Chen, Changzhi Sun, Jiaqing Liang, Yanghua Xiao, and Deqing Yang. ANALOGYKB: Unlocking analogical reasoning of language models with a million-scale knowledge base. In *Proceedings of ACL*, pp. 1249–1265, 2024.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabás Póczos, Russ R. Salakhutdinov, and Alexander Smola. Deep sets. In *Advances in NeurIPS*, volume 30, 2017.

Zexuan Zhong, Zhengxuan Wu, Christopher Manning, Christopher Potts, and Danqi Chen. MQuAKE: Assessing knowledge editing in language models via multi-hop questions. In *Proceedings of EMNLP*, pp. 15686–15702, 2023.

Jincheng Zhou, Yucheng Zhang, Jianfei Gao, Yangze Zhou, and Bruno Ribeiro. Double equivariance for inductive link prediction for both new nodes and new relation types. *arXiv preprint arXiv:2302.01313*, 2023.

Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal Xhonneux, and Jian Tang. Neural Bellman-Ford networks: A general graph neural network framework for link prediction. In *Advances in NeurIPS*, volume 34, pp. 29476–29490, 2021.

CONTENTS OF THE APPENDIX

- Appendix A defines the datasets, the rules they are built by, example episodes (Figure 4) and the controlled world.
- Appendix B gives the knowledge models, the operators, the objectives, training and the baselines.
- Appendix C proves Propositions 1 and 2.
- Appendix D gives the results behind Table 1 and Figures 2 and 3, grouped by the three findings, and the time split.
- Appendix E lists every preregistered criterion with its outcome, and every deviation from the protocols.

A DATASETS AND TRANSFORMATION RULES

A.1 TASK AND CONSTRUCTION RULES

Every dataset is mapped to one task. An episode gives $k \geq 1$ support pairs that share one relation, and a query entity; the model must predict the entity that stands in the same relation to the query. Knowledge about entities is fixed before relation learning, and the answer is retrieved among real entities. The rules that the claims depend on are:

R1 Hidden relation. Relation identifiers, names and role labels are used only to build and score episodes; they never enter the model input.

R2 Supports. Supports share the relation, use entities other than the query and, for transfer, come from other domains.

R3 Candidates. Candidates are the query's known components or a common universe of entities; they are never filtered by the gold.

R4 Held-out split. Test relations and test domains never occur in training episodes.

R5 Role swap. For each test query with two relations whose gold sets are disjoint, both episodes are built and scored jointly.

R6 Evidence type. A correct prediction counts as evidence of an unlearned correspondence only under deletion (E1: the query–gold fact, its inverse and every edge between them within the role family are deleted before the knowledge is fit), a counterfactual edit (E2), or a renamed or invented system (E3, E4). Results on frozen language-model states under none of these conditions are reported as access to known facts.

R7 Role families. A role family is a set of properties that realise one role in different domains. Families come only from connected components of AnalogyKB's analogous-property pairs (Yuan et al., 2024) and from Wikidata subproperty groups, without human judgement; a family is used across domains only if its members occur in at least two domains.

R8 Inferable stratum. Under deletion, an item is inferable if the query and the gold are still joined by a path of at most two edges; a claim of inference must beat personalised PageRank on this stratum.

Each dataset also ships with a manifest of source hashes, counts after each filter, audits of duplicate and inverse relations and of overlaps with training data, and a log of any rule changed after results were seen.

A.2 DERIVED DATASETS

Table 2: Evaluation suite, built under the rules above. Items are test episodes (split A / B where two splits exist); chance, the share of golds among the candidates, in parentheses. Of the time split's facts for queries the snapshot does not answer, 825 were verified as added to Wikidata after 2019 (Appendix D.6).

Suite	Frozen states	Why the answer is unobserved	Items	Candidates
Controlled world	64-d, fit to facts	4 of 12 domains held out	4 worlds	8–14 components
WikiTopics-E1	128-d graph	queried facts deleted (E1)	4,899 / 7,296	10–20k entities
MQuAKE-CF	128-d graph	one hop edited (E2)	300 edits	final hop's tails
SCAR roles	Qwen, Pythia	supports from other fields	235	19 components (0.06)
Renamed systems	Qwen, Pythia	renamed (E3); described edits (E2)	117	16 components (0.07)
Invented systems	Qwen, Pythia	invented (E4); described edits (E2)	360	9 components (0.11)
Nested invented	Qwen, Pythia	invented, two steps	360	9 per step (0.11)
Time split	128-d graph	not in the 2019 snapshot	18,835	200,000 entities

- **WikiTopics roles.** Built from WikiTopics (Zhou et al., 2023) with Wikidata5M labels (Wang et al., 2021).
  - Two held-out topic sets were drawn from the 49 admissible sets: split A holds out art, organisation, sport and taxonomy; split B holds out art, infrastructure, people and sport.
  - Test sets: A1 (seen relation, new query), A2 (unseen relation in unseen topics) and A3 (the same role family realised by a different property).
  - Deletion removes 58–68k of 604k facts. Between 20% and 47% of items are inferable.

- **MQuAKE-CF** (counterfactual). Built from the remastered CF-3k release (Zhong et al., 2023), mapped to Wikidata5M: 2,683 cases of 2, 3 and 4 hops, each with a locality partner, an unrelated chain whose answer must not change when the case is edited. 126 cases that depend on conflicting facts are dropped.
- **SCAR roles.** A node is a (system, concept) pair, and a SCAR mapping (Yuan et al., 2023) is an edge. A role is a connected component, excluding whole-system nodes, and it is used when at least three systems share it. Supports come from other fields (at most five). Candidates are the query system's known components. This gives 235 episodes over 104 query systems, 19 candidates on average, and 290 role-swap pairs.
- **Renamed systems.** For each SCAR episode whose query description names the gold and at least two other candidates, the system and all named candidates are replaced throughout the description by seeded pronounceable nonce words. Candidates whose name is contained in another candidate's name are dropped.
  - *Edit:* the gold's and one distractor's nonce names are swapped.
  - *Sham:* two non-candidate content words are swapped.
  - *Reorder:* sentences are reordered without changing any name–statement binding (gold first, distractor first, random).
- **Invented systems.** Every system is invented: a nonce name, a neutral kind noun and nine nonce-named components (controller and partial controller, boundary and partial boundary, gate and partial gate, member, and two fillers), whose roles are stated only relationally in ordinary English sentences, in shuffled order. Template set A (and its kind nouns) is used for supports and the 400 training systems, set B for the 200 test query systems, so queries share no sentence template or kind noun with supports. Names are unique across all systems. Each role has a full and a partial holder, which defines the graded order. In cross-mapped systems, a distinctive attribute sentence is carried by a filler in the query and by the role holders in the supports. Controls: names only, an edit that swaps the full holder's and a filler's names throughout the description, and a sham that swaps two other content words.
- **Nested invented systems.** Each system has a top level and a sub-level inside its controller, written with the same rules; a two-step episode asks for the controller of the query system and then for a role inside it, with separate supports for each step.

A.3 Example episodes

Figure 4 shows one episode of an invented system and one of a renamed system, as the model reads them.

A.4 Controlled knowledge world

The world has twelve domains with 30 systems each and 8–14 entities per system.

- **Roles.** Centre, member, boundary, gate and decoy. Each role is a bundle of properties, such as inside, at-middle, controls and linked for the centre. About 20% of systems split the centre into a structural and a control centre; they are used only for the test of choosing between two readings of a role (Appendix E).
- **Confounds.** A type shortcut picks a wrong entity in more than half of the held-out systems, and a salience shortcut (the entity with most facts) in 31% of systems.
- **The α axis.** A share α of role facts uses predicates shared across domains; the rest uses domain-specific tokens.
- **Worlds.** Four worlds are generated, $\alpha = 1$ with two seeds, $\alpha = 0.5$ and $\alpha = 0$; each result names the worlds it uses.
- **Knowledge.** A 64-d entity table with a relational decoder is trained to at least 99% recall on all facts, including those of held-out domains, and then frozen.
- **Relation learning.** Relation learning uses eight domains; four are held out entirely, including one nested pair (organism $\rightarrow$ cell) for composition.
- **Interventions.** Edits relearn only the edited entity rows, with the decoder frozen. A sham relearning with unchanged facts is run alongside.

a Invented system, controller role. One of three supports, written with template set A, pairs Zageto with its controller fubefe:

Zageto is a cluster. The sokesu can direct a few zaboras, but only with the fubefe's consent. The fubefe is striped black and white. The falafe covers one side of Zageto and leaves the rest open. The safiba is a crack in the veta that is almost always sealed. When the fubefe stops, every zabora stops as well. Zageto holds many zaboras. Every part of Zageto is enclosed by the veta. The ritada was added long after the rest. The roziza is covered in fine ridges. The dokuma is the only opening in the veta; whatever leaves Zageto passes through it.

The query, written with template set B, is Gekumu, and the candidates are its nine components:

Gekumu is a complex. Most of Gekumu consists of natadis. The vomavo is streaked with pale lines. The bazisi is striped black and white. Anything entering or leaving Gekumu must go through the tokine, a gap in the mufeva. The patova is streaked with pale lines. A narrow slit, the kibule, runs through the mufeva but rarely lets anything pass. Now and then the nofivi steers one or two natadis, always under the bazisi's authority. All the parts of Gekumu sit within the mufeva. Along one flank the lovufe shields Gekumu, but it does not go all the way around. No natadi moves on its own; the bazisi decides for them.

Gold **bazisi** (the full controller); partial holder nofivi, which defines the graded order. An edit swaps the names bazisi and patova, a filler, throughout the description, and the prediction must move to patova.

b Renamed system (SCAR). The supports come from other fields: (buy and sell, goods circulation rate), (chemistry, reaction speed), (crowd, contact frequency). The query system, dance, and its components are renamed:

Pomi is a captivating geka that encompasses toteko and simu. It involves a lefi of sope choreographed to a guteme, sometimes accompanied by nabe or pali. guteme is an leka of Pomi as it allows the movements to flow seamlessly, highlighting the rema and refinement of each Pomi step. [. . .]

Gold **guteme** (rhythm); distractor geka (art form). The edit swaps guteme and geka throughout the description, and the prediction must move to geka.

Figure 4: Example episodes. The model sees only the descriptions and the support pairs; names in bold are the golds. Relation and role names are never shown.

B Implementation details

Knowledge models. In the controlled world and on Wikidata, the knowledge model that yields the knowledge-graph states has four parts:

- an entity table E (64-d in the controlled world, 128-d on Wikidata);
- a predicate table;
- an MLP decoder $f(\cdot \mid x, p)$ that scores every entity (1-to-N);
- an auxiliary head that predicts which predicates an entity is the subject of.

The model is trained on every fact and its reciprocal. On Wikidata, training stops when filtered hits@1 (the share of facts whose object is ranked first once the other true objects are removed) on a fixed sample of 20k training facts reaches 0.90, or after 60 epochs; the same rule applies to every graph. The table and the decoder are then frozen. The transition state $m_p(x)$ is the decoder's expected entity state, $\sum_y f(y \mid x, p)\, E(y)$.

Language-model states. A component's state is the mean hidden state over the tokens of its name in the phrase "In {system}, the {component}", optionally preceded by the system's description. We use Qwen2.5-7B (layer 21) and Pythia-6.9B (layer 8). Before any relation computation, states are centred by the mean over all component states and normalised. The layers were fixed by a preregistered rule, the development score of an earlier SCAR experiment, before any result reported here.

Operators. Table 3 lists every operator by what it reads and what it outputs. Learned operators share one reader. Each support pair is encoded from $[a, b, b-a]$ by an MLP after a learned projection of the states, and the codes are mean-pooled into h.

Objectives and training.

- **Latent variable (Pred+z).** The attention scorer outputs four logits per item, giving four states. For the learned centred offset, four relation maps start at the identity plus $\mathcal{N}(0, 0.02^2)$ noise. The

Table 3: Operators of R-JEPA and the results that use them. Attention scores each item by an MLP of the projected query, h and the item, and $\hat{z}$ is the softmax-weighted mean of the item states, a convex combination.

Operator	Reads	Output	Used for
centred offset (no learning)	role vector h of the supports and the mean state $\bar{m}_q$ of the query system	$\bar{m}_q + h$	SCAR, renamed and invented systems, controlled world
predicate matching (no learning)	the stored predicate $\hat{p}$ that best explains the supports	$m_{\hat{p}}(q)$	deleted facts, MQuAKE, time split
component attention	states of the query's components	convex combination	controlled world
neighbour attention	states of the query's graph neighbours and the linking edges	convex combination	graph states, facts kept
transition attention	decoder transitions $m_p(q)$ for the predicates of the query's and the supports' topics, in both directions	convex combination	controlled world, MQuAKE
with support fit	also each predicate's fit $\frac{1}{k}\sum_i \log f(b_i \mid a_i, p)$	convex combination	deleted facts
with residual	adds a residual MLP	unconstrained	controlled world reuse
learned centred offset	projection W to 256 dimensions and a relation map initialised at the identity	projected state	renamed and invented systems, SCAR
residual transition	$\mathrm{normalise}(z + h + \mathrm{MLP}([z, h]))$, last layer initialised at zero	unconstrained	nested invented systems
MLP operator	query state and h	unconstrained	time split, held-out relations

head closest to the sampled gold takes weight 0.95 and the others share 0.05, and latent rollouts on MQuAKE enumerate all 4^{hops} head sequences.

- **Withheld negatives.** Cls_n samples n wrong candidates per item and step with a separate random generator, so its batches equal those of Cls with the same seed. Cls_b uses one sampled gold per item as the candidates of every other item.
- **Stopped target.** With the learned centred offset, the Pred target is the gold's state under an EMA copy of W (momentum 0.99). Some tables also report Pred+Cls, the weighted sum $\mathcal{L}_{\text{Pred}} + 0.1\,\mathcal{L}_{\text{Cls}}$.
- **Settings.** Adam with learning rate 10^{-3}, batch 64 and 6,000 steps on graphs, and AdamW with batch 32 and 2,000 steps for the learned centred offset. Model selection uses the MRR on development items of training domains only. On real data, half of each batch consists of role swap pairs.
- **Checks.** Each new training script reproduces its reference. Cls_n with all eight negatives reproduces the Cls run on invented systems (Pythia, seed 0) exactly (top-1 0.753), and the saved graph models re-evaluate to their recorded top-1.

Training data for learned operators on language-model states. SCAR is too small to train on. We therefore train in three ways:

- on Wikidata facts written in the same template, evaluated zero-shot on SCAR;
- on renamed systems, with each training system renamed four times so that names cannot be memorised, and with four field folds;
- on invented systems, with 400 training systems whose sentence templates and kind nouns differ from the test queries'.

Language-model baselines.

- **In context.** The description (for renamed and invented systems) is followed by one "{system}: {component}" line per support and by "{query system}:". Each candidate is scored by its mean token log-probability.
- **Task vectors** (Hendel et al., 2023). The hidden state at the last support position is patched into a zero-shot query at layer $\mathrm{round}(L/3)$ of the model's L layers.

- **Function vectors** (Todd et al., 2024). Attention heads are ranked by average indirect effect on function-vector tasks. The vector sums the mean outputs of the top 10 heads. The layer at which it is added is chosen on those tasks (Qwen layer 18). On Pythia the sweep peaks at 0.16 (layer 11) and the vector does not change zero-shot accuracy on those tasks, so the method does not validate there.
- **ROME** (Meng et al., 2022). We use our implementation at layer 5, with the key covariance estimated on 2,000 Wikidata5M descriptions. An edit takes 20 optimisation steps with a KL and a norm penalty and then makes a rank-one update. A role edit transfers up to three statements of the gold to the distractor and vice versa, applied sequentially. The sham re-asserts each concept's own statements.

Parameter-free baselines on graphs.

- **Neighbour mean:** the mean state of the query's neighbours.
- **Personalised PageRank:** restart probability 0.15, seeded at the query.
- **Decoder given the true relation (oracle):** the frozen decoder applied with the true predicate.

Compute. All experiments ran on a single 24 GB GPU.

C PROOFS

Proposition 1. Write $x_c = E(c) - \bar{m}_q = \lambda_c(r_{\rho(c)} + \epsilon_c)$ with $\lambda_c > 0$. By assumption each support difference is $\beta_i(r_\rho + \epsilon_i)$ with $\beta_i > 0$, so

$$h = \frac{1}{k}\sum_i \beta_i(r_\rho + \epsilon_i) = \bar{\beta}\,(r_\rho + \bar{\epsilon}), \qquad \bar{\beta} = \frac{1}{k}\sum_i \beta_i, \quad \bar{\epsilon} = \frac{\sum_i \beta_i \epsilon_i}{\sum_i \beta_i},$$

and $\|\bar{\epsilon}\| \le \delta$ because $\bar{\epsilon}$ is a convex combination of vectors of norm at most δ. The centred offset ranks candidates by $\cos(x_c, h)$. The positive factors λ_c, $\bar{\beta}$ and $\|h\|$ do not change this ranking, so it equals the ranking by

$$\sigma(c) = \frac{\langle r_{\rho(c)} + \epsilon_c,\ r_\rho + \bar{\epsilon}\rangle}{\|r_{\rho(c)} + \epsilon_c\|}.$$

Holder of ρ. Let $\rho(c) = \rho$. The numerator is $1 + \langle r_\rho, \bar{\epsilon}\rangle + \langle \epsilon_c, r_\rho\rangle + \langle \epsilon_c, \bar{\epsilon}\rangle \ge 1 - 2\delta - \delta^2$, and the denominator is at most $1 + \delta$. The numerator is positive because $\gamma \le 2$ and $\gamma > 6\delta$ give $\delta < 1/3$. Hence $\sigma(c) \ge (1 - 2\delta - \delta^2)/(1 + \delta)$.

Any other candidate. Let $\rho(c') \ne \rho$ and let N be the numerator. Then $N \le 1 - \gamma + 2\delta + \delta^2$, and the denominator is at least $1 - \delta > 0$.

- If $N \ge 0$, then $\sigma(c') \le N/(1 - \delta)$.
- If $N < 0$, then $\sigma(c') < 0 < \sigma(c)$.

In the first case, $\sigma(c) > \sigma(c')$ follows from

$$(1 - 2\delta - \delta^2)(1 - \delta) > (1 - \gamma + 2\delta + \delta^2)(1 + \delta).$$

Expanding, the left side is $1 - 3\delta + \delta^2 + \delta^3$. The right side is $(1 - \gamma)(1 + \delta) + 2\delta + 3\delta^2 + \delta^3$. Their difference is $\gamma(1 + \delta) - 6\delta - 2\delta^2$. This difference is positive whenever $\gamma > (6\delta + 2\delta^2)/(1 + \delta)$, and this bound is at most 6δ. So every holder of ρ outranks every other candidate. □

Remarks. The condition refers to the geometry of the centred states: to role directions and their separation. An invertible linear map of the state space preserves which entity holds which role, but it can shrink γ relative to δ. A learned operator can undo such a map; a fixed operator cannot. On renamed systems, Qwen layer 7 has less of this geometry than layer 21, and the centred offset drops from 0.34 to 0.13 (Appendix D.4).

Proposition 2. Let $e_c = E(c)$, with $\|e_c\| = 1$ and $e_c \in U$.

Classification. $s_c = \cos(\hat{z}, e_c) = \langle u, e_c\rangle = \langle \Pi_U u, e_c\rangle$, so $\mathcal{L}_{\text{Cls}}$ is a function of $\Pi_U u$ alone. Fix $v = \Pi_U u$ with $\|v\| < 1$. The unit vectors with this projection are $v + w$ with $w \in U^\perp$ and

$\|w\|^2 = 1 - \|v\|^2$. This set is a sphere of dimension $\dim U^\perp - 1 = d - n_U - 1$, and all of its points attain the same $\mathcal{L}_{\mathrm{Cls}}$.

Latent prediction. $\mathcal{L}_{\mathrm{Pred}} = 1 - \langle u, e_{y^\star} \rangle$ is minimised, over unit vectors, only at $u = e_{y^\star}$. For unit vectors, $\|u - e_{y^\star}\|^2 = 2 - 2\langle u, e_{y^\star} \rangle = 2\mathcal{L}_{\mathrm{Pred}}$. The Cauchy–Schwarz inequality then gives $|\langle u - e_{y^\star}, w \rangle| \le \sqrt{2\mathcal{L}_{\mathrm{Pred}}}\,\|w\|$. □

Consequence for reuse. Suppose the next step scores its candidates along directions w that have a component outside U. Under $\mathcal{L}_{\mathrm{Cls}}$, the contribution $\langle u, \Pi_{U^\perp} w \rangle$ is not constrained by training. It can take any value in $\pm\sqrt{1 - \|\Pi_U u\|^2}\,\|\Pi_{U^\perp} w\|$. Under $\mathcal{L}_{\mathrm{Pred}}$, it is bounded by the training loss. If the operator's output is restricted to the convex hull of known states, u is determined by the mixture weights. The architecture then removes the free component, whatever the objective.

Sampled candidates. If classification is trained on a sampled subset $\mathcal{S} \subset \mathcal{C}(q)$ that contains the golds, $\mathcal{L}_{\mathrm{Cls}}$ is the same function with $\mathcal{C}(q)$ replaced by $\mathcal{S}$, so Proposition 2 applies with $U_{\mathcal{S}} = \mathrm{span}\{E(c) : c \in \mathcal{S}\}$: the loss depends on u only through $\Pi_{U_{\mathcal{S}}} u$. Any component of u that separates the gold from a candidate outside $\mathcal{S}$ and is orthogonal to $U_{\mathcal{S}}$ receives no gradient from that step.

D Additional results

This appendix gives the numbers behind Table 1 and the figures, grouped by the three findings of the introduction. Each table is preceded by what it measures and what it shows. All tables are generated from the result files by the scripts of the supplementary material, and learned operators use three seeds unless noted.

D.1 How Table 1 was built

Each row of Table 1 uses one operator and one evaluation set:

- deleted facts: transition attention with support fit on test set A2 (unseen relations in unseen topics) of each split;
- roles across fields: the centred offset on all 235 SCAR episodes; the learned columns are the learned centred offset trained on Wikidata facts and applied zero-shot (three seeds);
- renamed systems: every column on the 85 systems of held-out fields with enough supports, in the states used to train the learned centred offset (four folds of held-out fields); Appendix D.3 reports all 117 systems;
- invented systems: the learned centred offset trained on the 400 training systems;
- controlled world: component attention on the held-out domains; graph states: neighbour attention with all facts kept, test set A2;
- reuse: transition attention with residual on the held-out nested pair of the controlled world (organism → cell; mean of two worlds), the residual transition on nested invented systems, and transition attention on the 884 three-hop MQuAKE chains. The parameter-free columns apply a fixed step twice: in the controlled world the centred offset and then the offset of the second role (baseline: offset twice), ranking all 322–332 entities of the inner domain, of which a role can have several holders (chance 0.007); on invented systems the centred offset twice (baseline: prototype of the second role); on MQuAKE predicate matching at every hop (baseline: offset);
- renamed systems, Qwen layer 7: the renamed-systems setup in the states of Qwen layer 7;
- roles across predicates: deletion (E1), test set A3 of split A, with predicate matching, transition attention with support fit and personalised PageRank.

D.2 The predicted state as a world model state (first finding)

Edits and reuse on a real graph. Table 4 gives the MQuAKE edits and rollouts of the learned transition attention. Its sham control changes 0.17 to 0.21 of all chains but only 0.01 to 0.03 of the chains that were correct before the edit. Predicate matching composes better than the learned operators because it names the true relation in 0.95 of hops and the chain facts are stored, and the offset baseline stays near chance.

Table 4: MQuAKE with knowledge-graph states. Top: edits (Pred mean ± s.d. of three seeds, Cls seed 0); follow among chains whose rollout was correct and whose edited fact was learned; sham over all chains and over chains correct before the edit. Bottom: final-hop top-1 for 2 / 3 / 4 hops (three seeds): latent (predicted state fed back), grounded (retrieved entity re-embedded), teacher (true intermediate entity). Pred+z: latent variable, each candidate scored by its best head, all head sequences. Predicate matching and offset have no parameters (one run).

operator	follow	locality	sham, all	sham, base correct
transition attention, Pred	0.780±0.031	0.966±0.002	0.170±0.026	0.033±0.010
transition attention, Cls	0.738	0.943	0.207	0.014
transition attention	**latent 2/3/4**	**grounded 2/3/4**	**teacher 2/3/4**	
Pred	0.57 / 0.27 / 0.54	0.65 / 0.58 / 0.66	0.66 / 0.62 / 0.76	
Cls	0.55 / 0.26 / 0.47	0.64 / 0.37 / 0.58	0.65 / 0.55 / 0.65	
Pred+z	0.56 / 0.30 / 0.55	0.67 / 0.57 / 0.73	0.68 / 0.63 / 0.85	
predicate matching (no learning)	0.83 / 0.48 / 0.70	0.90 / 0.69 / 0.91	0.91 / 0.78 / 1.00	
offset (baseline)	0.04 / 0.04 / 0.02	0.03 / 0.05 / 0.05	0.21 / 0.13 / 0.52	

Reuse in language-model states. Table 5 gives two-step reuse on nested invented systems. Latent prediction keeps the predicted state closer to the true one (lower state error), but classification is more accurate after the rollout. The second step ranks the components of the true inner system whatever the first step retrieved, so a prototype that ignores the first step already reaches 0.91 (Qwen) and 0.30 (Pythia).

Table 5: Two-step reuse on nested invented systems (360 episodes; learned rows: the residual transition, three seeds). Step 1 predicts the controller of the system, step 2 a role inside it, from the predicted state (rollout) or from the true controller (teacher). State error: $1 - \cos$ to the true controller after step 1.

model	operator	step 1	teacher	rollout	state error
Qwen	centred offset	0.914	0.797	0.714	0.485
Qwen	Pred	1.000	0.946	0.955	0.382
Qwen	Cls	1.000	1.000	0.999	0.487
Pythia	centred offset	0.744	0.197	0.194	0.582
Pythia	Pred	0.996	0.518	0.479	0.537
Pythia	Cls	1.000	0.660	0.588	0.607

D.3 Where relations transfer without learning (second finding)

Role geometry. Table 6 gives κ for each state space. Values compare only within a family of states, because graph states must single out one entity among 10 to 20 thousand while language model states choose among about 16 components.

Role consistency across language models. Table 7 gives κ and the centred offset for seven language models at seven depths each, built with the same template. Six of the seven models show a positive correlation within the model, and the last layer is the main exception. Gemma's residual stream exceeds the float16 range, so its states and those of Qwen3-8B were stored in float32.

Roles across scientific fields. Table 8 shows that the centred offset uses the supports. Both answers of a role swap pair are right in 0.44 and 0.39 of pairs, and shuffling the supports lowers top-1 to 0.15 and 0.11.

Renamed systems. Table 9 lists the controls on all 117 renamed systems. For Qwen, accuracy falls to chance without the description, predictions follow edits much more often than sham edits, and no reordering moves a prediction to the distractor. Pythia transfers and follows edits less, below the registered threshold (follow ≥ 0.5, Appendix E), and its transfer depends on how the states are

Table 6: Role consistency κ (Section 5.4): mean cosine between same-role support differences from different domains minus that between different roles. Comparable within a family of states only.

state space	κ
controlled world, $\alpha = 1$ (seed 0)	0.305
controlled world, $\alpha = 1$ (seed 1)	0.317
controlled world, $\alpha = 0.5$	0.173
controlled world, $\alpha = 0$	0.156
Qwen layer 7, described systems	0.123
Qwen layer 21, described systems	0.164
Qwen layer 21, names only	0.150
Pythia layer 8, names only	0.105
knowledge-graph states, split A (A2 supports)	0.236
knowledge-graph states, split B (A2 supports)	0.202

Table 7: Role consistency κ and the centred offset (SCAR top-1) at seven depths (25%–100%) of each language model; SCAR states as in Table 8. Over all 49 points, Spearman 0.45 [0.19, 0.65] (preregistered, points resampled). Resampling models, whose seven layers enter together, gives [0.17, 0.67], and leaving one model out gives 0.35 to 0.54. Best-κ layer: top-1 at the layer of highest κ against the model's median.

model	κ (range)	top-1 (range)	within-model Spearman	best-κ layer
Qwen2.5-7B	0.110–0.150	0.35–0.57	+0.33	0.57 vs 0.56
Pythia-6.9B	0.105–0.135	0.43–0.52	-0.41	0.50 vs 0.51
Llama-3.2-3B	0.127–0.155	0.46–0.57	+0.71	0.57 vs 0.55
Gemma-3-4B	0.094–0.140	0.25–0.47	+0.71	0.44 vs 0.34
OLMo-2-1B	0.104–0.151	0.46–0.57	+0.74	0.54 vs 0.54
Qwen3-4B	0.110–0.145	0.46–0.54	+0.50	0.54 vs 0.53
Qwen3-8B	0.109–0.135	0.36–0.54	+0.50	0.52 vs 0.52

Table 8: SCAR roles (235 episodes, 290 role-swap pairs; chance 0.062). Swap: both episodes of a role-swap pair correct. The language models' own mechanisms are in Appendix D.3.

	Qwen2.5-7B		Pythia-6.9B	
	top-1	swap	top-1	swap
centred offset (no learning)	0.566	0.441	0.506	0.386
prototype	0.519	0.352	0.485	0.341
centred offset, shuffled supports	0.145	–	0.111	–

centred and on the system set (0.16 on these 117 systems, 0.08 on the 85 systems and states of Appendix D.4), while Qwen's does not (0.34 in both).

Table 9: Renamed (nonce) systems: controls for the centred offset. Follow, stay and to-distractor are computed on the items the operator gets right before the intervention; reordering: three sentence orders (range over orders; to distractor: the largest share). In-context answers are in Appendix D.3.

	Qwen2.5-7B	Pythia-6.9B
systems	117	117
chance	0.071	0.071
top-1	0.342	0.162
top-1, names only	0.077	0.068
follows the edit	0.650	0.368
sham changes	0.085	0.034
stays under reordering	0.72 – 0.93	0.47 – 0.74
to distractor under reordering	0.000	0.158

Invented systems. Table 10 reports the systems invented for the test. Operators trained on other invented systems are strongest, except that latent prediction trails the model in context in Pythia (0.66 against 0.69). Among them classification leads on top-1, and latent prediction keeps the graded order of full and partial role holders better in Qwen (0.95 against 0.88).

Table 10: Invented systems (360 test episodes, 9 candidates, chance 0.111). Learned: the learned centred offset, three seeds, trained on 400 training systems. Graded: gold scored above the partial holder, and both above the mean of the others. Follow / sham: description edits. Variants of the objectives are in Appendix D.5.

	Qwen2.5-7B, layer 21			Pythia-6.9B, layer 8		
method	top-1	graded	follow / sham	top-1	graded	follow / sham
prototype	0.869	0.789	–	0.406	0.225	–
centred offset (no learning)	0.875	0.794	0.94 / 0.04	0.411	0.228	0.70 / 0.04
model in context	0.925	0.772	–	0.686	0.531	–
centred offset, names only	0.103	–	–	0.108	–	–
Pred (EMA target)	0.981	0.955	1.00 / 0.01	0.659	0.480	0.98 / 0.04
Cls	0.991	0.884	1.00 / 0.01	0.743	0.569	0.91 / 0.04

The language models' own mechanisms. Table 11 compares the centred offset with the models' own use of the examples, answering in context (ICL) and adding a task vector (TV) or a function vector (FV) to the query's forward pass. Task and function vectors stay near chance on renamed systems. The model in context leads on invented systems, where it reads the full support descriptions, and over two steps, which task and function vectors do not define.

Table 11: The language models' own mechanisms on the SCAR, renamed (all 117) and invented items: the model answering in context (ICL), task vectors (TV) and function vectors (FV), next to the centred offset. n/v: the function-vector method does not validate on Pythia. Two steps: ICL chains its own first answer (rollout) or is given the true controller (teacher); task and function vectors do not define a two-step rollout.

setting	chance	ICL	TV	FV	parameter-free R-JEPA
roles across fields, Qwen	0.06	0.40	0.08	0.05	0.57
roles across fields, Pythia	0.06	0.23	0.06	n/v	0.51
renamed systems, Qwen	0.07	0.15	0.04	0.05	0.34
renamed systems, Pythia	0.07	0.09	0.07	n/v	0.16
invented systems, Qwen	0.11	0.93	0.16	0.41	0.88
invented systems, Pythia	0.11	0.69	0.35	n/v	0.41
two steps, invented, Qwen	0.11	0.89 / 0.95	–	–	0.71 / 0.80
two steps, invented, Pythia	0.11	0.70 / 0.74	–	–	0.19 / 0.20

D.4 Where the operator must be learned (second finding)

Controlled world. Table 12 gives every learned operator. Component attention lifts transfer from the best reference without learning (0.37 to 0.67) to 0.72 to 0.99, with latent prediction at least as high as classification, while transition attention gains less where domains share no predicate words.

Wikidata graphs. Table 13 covers both graph conditions. With facts kept, neighbour attention reaches 0.69 to 0.73 on split A and 0.43 to 0.51 on split B, and the objectives trade places between splits. The inferable column is the stratum where query and answer are still joined by a path of at most two edges.

Language-model states. Table 14 tests learning on renamed systems. At Qwen layers 21 and 7 no learned operator beats the centred offset beyond seed variation, while at Pythia layer 8, where the centred offset is near chance in these states, learned operators reach 0.20 to 0.23.

Table 12: Controlled world, held-out domains (3 seeds per arm). Baseline: best parameter-free reference.

		component attention			transition attention		
world	baseline	Pred	Cls	Pred+Cls	Pred	Cls	Pred+Cls
$\alpha = 1$, seed 0	0.673	0.970	0.946	0.971	0.964	0.922	0.959
$\alpha = 1$, seed 1	0.640	0.985	0.954	0.980	0.900	0.860	0.883
$\alpha = 0.5$	0.448	0.886	0.867	0.882	0.699	0.654	0.712
$\alpha = 0$	0.373	0.756	0.717	0.749	0.509	0.511	0.540

Table 13: WikiTopics with knowledge-graph states (learned models: mean of three seeds). Known: all facts kept. E1: query–gold facts deleted before the knowledge is fit; inferable = still joined by a path of length ≤ 2 (Appendix A). Top-1, split A / split B.

	known A2	known A3	E1 A2	E1 A2 inferable	E1 A3
decoder given the true relation	–	–	0.29 / 0.27	0.18 / 0.21	–
predicate matching	0.99 / 0.99	0.64 / 0.41	0.29 / 0.26	0.18 / 0.21	0.01 / 0.01
neighbour attention, Pred	0.69 / 0.51	0.71 / 0.37	–	–	–
neighbour attention, Cls	0.72 / 0.43	0.61 / 0.40	–	–	–
neighbour attention, Pred+Cls	0.73 / 0.45	0.70 / 0.37	–	–	–
transition attention with support fit, Pred	–	–	0.20 / 0.18	0.12 / 0.10	0.03 / 0.02
transition attention with support fit, Cls	–	–	0.21 / 0.19	0.12 / 0.11	0.03 / 0.01
neighbour mean	0.38 / 0.26	0.63 / 0.44	0.00 / 0.00	0.00 / 0.01	0.00 / 0.00
prototype	0.18 / 0.12	0.01 / 0.01	0.18 / 0.12	0.11 / 0.06	0.01 / 0.00
offset	0.17 / 0.22	0.00 / 0.12	0.16 / 0.20	0.10 / 0.26	0.00 / 0.06
personalised PageRank	0.27 / 0.23	0.27 / 0.23	0.13 / 0.06	0.00 / 0.01	0.22 / 0.04

Table 14: Learning on language-model states (85 renamed systems from fields held out from training; 3 seeds, mean$\pm$sd). Learned centred offset: a learned projection to 256 dimensions plus a relation map, trained from scratch; Pred regresses onto an EMA copy of the projected target. Layers 21 and 7: Qwen.

states	model	top-1	follows edit
layer 21	parameter-free	0.341	0.724
layer 21	learned centred offset, Pred (EMA target)	0.282$\pm$0.010	0.569
layer 21	learned centred offset, Cls	0.314$\pm$0.028	0.738
layer 7	parameter-free	0.129	0.091
layer 7	learned centred offset, Pred (EMA target)	0.129$\pm$0.017	0.244
layer 7	learned centred offset, Cls	0.133$\pm$0.034	0.369
Pythia, layer 8	parameter-free	0.082	0.286
Pythia, layer 8	learned centred offset, Pred (EMA target)	0.196$\pm$0.006	0.321
Pythia, layer 8	learned centred offset, Cls	0.231$\pm$0.006	0.524

D.5 When predicting the state beats classifying (third finding)

Scarce negatives. Table 15 is the full result behind Figure 3b, with the evaluation fixed and only the negatives changed. Negatives drawn from the batch lead latent prediction by 0.02 on split A and trail it by 0.14 on split B, so they are no reliable substitute. The intervals in the table resample query entities of seed-averaged differences. Resampling seeds and entities together (Table 16) keeps Pred $-$ Cls_1 above zero on both splits, and it is positive in every seed. On split B, Pred also stays ahead of Cls_{16}, Cls and Cls_b. On split A, Cls_{16}, Cls and Cls_b are within seed variation of Pred, as one seed of Cls leads by 0.13 and the others trail by up to 0.06.

Ranking every answer. On graph queries with several answers, Pred ranks the whole answer set higher than Cls in R-precision, the share of the answers among the top $|Y^\star(q)|$ retrievals (0.76 against 0.61 on split A, 0.63 against 0.43 on split B). Classification that averages its loss over the answers (Cls_avg) reaches the same level (0.76 and 0.60), so the difference comes from how Cls pools its positives.

Table 15: Classification with withheld negatives against latent prediction (three seeds; primary metric of each setting). Cls_n: n negatives sampled per step from the candidate set; Cls_b: the other items' targets in the batch only. Differences: paired bootstrap 95% intervals over query entities (graph), chains (MQuAKE) or query systems (invented).

setting	Pred	Cls_1	Cls_4	Cls_{16}	Cls	Cls_b	Pred − Cls_1	Pred − Cls_b
graph A, A2	0.693	0.560	0.629	0.679	0.716	0.714	$+0.13$ [$+0.12$, $+0.14$]	-0.02 [-0.03, -0.01]
graph B, A2	0.509	0.391	0.392	0.413	0.428	0.371	$+0.12$ [$+0.11$, $+0.13$]	$+0.14$ [$+0.13$, $+0.15$]
MQuAKE, one step	0.675	0.701	0.676	0.652	0.618	0.697	-0.03 [-0.04, -0.02]	-0.02 [-0.03, -0.02]
invented, Qwen	0.981	0.982	0.990	–	0.991	0.988	-0.00 [-0.01, $+0.01$]	-0.01 [-0.02, $+0.00$]
invented, Pythia	0.659	0.756	0.749	–	0.743	0.636	-0.10 [-0.12, -0.07]	$+0.02$ [$+0.01$, $+0.04$]

Table 16: Seed robustness of the graph comparisons in Table 15 (A2 top-1). Seed s: Pred of seed s against Cls of seed s. Interval: 95% bootstrap that resamples the three training seeds of each objective and the query entities together.

	graph A				graph B			
difference	seed 0	seed 1	seed 2	interval	seed 0	seed 1	seed 2	interval
Pred − Cls_1	$+0.18$	$+0.13$	$+0.09$	[$+0.07$, $+0.19$]	$+0.15$	$+0.12$	$+0.08$	[$+0.06$, $+0.18$]
Pred − Cls_4	$+0.08$	$+0.12$	-0.01	[-0.00, $+0.13$]	$+0.16$	$+0.11$	$+0.08$	[$+0.06$, $+0.18$]
Pred − Cls_{16}	$+0.04$	$+0.03$	-0.03	[-0.05, $+0.07$]	$+0.17$	$+0.09$	$+0.03$	[$+0.04$, $+0.16$]
Pred − Cls	$+0.06$	$+0.00$	-0.13	[-0.09, $+0.04$]	$+0.20$	$+0.05$	-0.01	[$+0.02$, $+0.15$]
Pred − Cls_b	$+0.01$	-0.02	-0.06	[-0.09, $+0.03$]	$+0.22$	$+0.13$	$+0.07$	[$+0.08$, $+0.20$]

Relation novelty and the number of negatives. Table 17 repeats the comparison on the knowledge of the time split (Appendix D.6), with 93 relations held out from operator training. Classification with 16 or 256 negatives leads latent prediction on seen and held out relations alike, latent prediction stays within 0.02 of classification over all entities, and classification with one negative trails on seen relations and ties on held out ones. With Figure 3b, the number of negatives that suffices depends on the graph, 16 on this graph and on split A, and none on split B.

Table 17: Latent prediction (Pred) against classification over all 200,000 entities (Cls) or with n sampled negatives (Cls_n), for relations seen in operator training and for 93 relations held out from it (time-split knowledge of Appendix D.6; recorded facts of held-out subjects; three seeds). Top: MRR. Bottom: Pred − Cls_n ± half the 95% interval (paired bootstrap over subjects).

relations	facts	Pred	Cls	Cls_1	Cls_{16}	Cls_{256}
seen	12,296	0.697	0.694	0.643	0.749	0.761
unseen	7,704	0.748	0.730	0.745	0.791	0.789
Pred − Cls_n, seen		–	$+0.010 \pm 0.004$	$+0.070 \pm 0.006$	-0.044 ± 0.004	-0.056 ± 0.003
Pred − Cls_n, unseen		–	$+0.019 \pm 0.004$	$+0.006 \pm 0.006$	-0.035 ± 0.006	-0.034 ± 0.006

A latent variable. Table 18 gives Pred+z in every setting. It loses where every entity of a topic is a candidate, gains where candidates are typed by relation (MQuAKE) and changes nothing where each role has one holder, and its heads cover several answers where several exist.

The JEPA objective as the entity encoder. Table 19 trains the controlled world's entity states themselves. The JEPA objective does not fit the facts (at most 0.79 against 0.99), so the preregistered comparison could not be run. In exploratory measures its states have no role geometry, and the centred offset falls to chance.

Table 18: Latent variable (Pred+z: four heads, relaxed winner-take-all, each candidate scored by its best head) against Pred and Cls (three seeds). Heads hit ≥ 2 golds: share of multi-answer A2 items whose heads' own top-1 retrievals are two or more distinct golds. MQuAKE rows: all 2,683 chains (teacher: final hop from the true input). Head cos: mean cosine between the heads' states.

setting	Pred	Cls	Pred+z	Pred+z − Pred	mechanism
graph A, A2 top-1	0.693	0.716	0.601	-0.09 [-0.10, -0.08]	heads hit ≥ 2 golds 0.79
graph B, A2 top-1	0.509	0.428	0.408	-0.10 [-0.11, -0.09]	heads hit ≥ 2 golds 0.50
MQuAKE, teacher	0.675	0.618	0.717	$+0.04$ [$+0.03$, $+0.05$]	Pred+z − Cls $+0.10$ [$+0.09$, $+0.11$]
MQuAKE, grounded	0.630	0.530	0.656	$+0.03$ [$+0.02$, $+0.03$]	Pred+z − Cls $+0.13$ [$+0.11$, $+0.14$]
MQuAKE, latent, 3 hops	0.267	0.263	0.297	$+0.03$ [$+0.02$, $+0.04$]	–
invented, Qwen	0.981	0.991	0.984	$+0.00$ [$+0.00$, $+0.01$]	head cos 0.98
invented, Pythia	0.659	0.743	0.669	$+0.01$ [$+0.00$, $+0.02$]	head cos 0.99

Table 19: Training the entity states of the controlled world ($\alpha = 0$, seed 0) with classification over entities (Cls) or with the JEPA objective (JEPA: regression of the decoded state onto an EMA copy of the object's state, with a variance regulariser; at most 1,000 epochs). JEPA never reached the preregistered fit of 0.99, so the comparison is void by protocol and the columns are exploratory. Facts: filtered hits@1 on the training facts; probe: balanced accuracy of a role probe on held-out domains; centred offset: its top-1 on these states.

encoder objective	facts	κ	role probe	centred offset
Cls	0.990	0.156	0.602	0.496
JEPA	0.713	0.003	0.303	0.225
JEPA, no regulariser	0.791	0.001	0.375	0.221

D.6 A TIME SPLIT OF WIKIDATA (FIRST AND THIRD FINDINGS)

The knowledge model is fit to Wikidata5M (Wang et al., 2021), built from a 2019 Wikidata dump, restricted to its 200,000 entities of highest degree (2.24 million facts, 577 relations), and then frozen. We read the current statements of the same entities from the Wikidata Query Service (September 2026) and keep the facts between them, under a relation of the snapshot, that the snapshot records in neither direction; ontology relations such as *instance of* are not used as queries. A fact is a *later answer* if the snapshot answers its query (s, p) with another object, and a *new query* otherwise; a new query has a *new subject type* if none of the subject's types (its *instance of* objects) is the type of any subject the snapshot records for p. Subjects are split 90/5/5. Operators are trained on the recorded facts of training subjects, selected on development subjects, and never see a later fact; each query comes with five recorded pairs of p as supports, and ranking is over all 200,000 entities with the query's other answers removed. The learned operator, the MLP operator of Table 3, reads the supports and maps the subject's state to a predicted state without a decoder, so that the objective alone shapes the prediction.

A fact missing from the snapshot was not necessarily added after it: 20 of 50 random missing facts were already on the subject's Wikidata page at the end of July 2019, as Wikidata5M keeps only part of its dump. We therefore test the claim that a fact was *added later* on a verified subset: for a random sample of 600 new-query test items, for all 723 new-query items of the new subject type, and for random samples of 400 later-answer items of test and of training subjects, we read the subject's last revision before August 2019 and keep the items whose fact was absent (57–68% of each set). The 825 verified new-query facts therefore include all 492 verified facts of the new subject type.

Table 20 gives the results and Table 21 the registered tests (H1–H4) with their outcomes. Predicate matching ranks each fact where the decoder given the true relation does (the same rank for 99.6% of new-query facts), on the full sets, and matches or exceeds it on the verified subsets (0.317 on 825 new-query facts added after 2019), and also for subjects of a type the relation was never recorded for (0.345; verified 0.276), where learned operators are 0.10–0.14 below on all such facts and 0.04–0.08 on the verified ones. The learned operators perform alike under both objectives on every set (new queries: 0.370 and 0.369). Classification over all entities was trained with the later answers of its training queries among its negatives, and its ranking of those answers matches that of latent predic-

tion for training and held-out subjects alike (difference in differences 0.001 on the full sets; 0.014 [−0.002, 0.031] on the 467 verified items). This is the regime of Proposition 2 in which classification sees every candidate; the gap of Section 5.4 appears when it sees few. Classification with 16 sampled negatives is the best learned operator on every set. The intervals resample subjects. Resampling the 323 relations instead, so that all facts of a relation enter together, changes no conclusion. Predicate matching stays within 0.01 of the decoder given the true relation on every relation, the two learned objectives still tie (Pred − Cls $+0.001$ [-0.005, $+0.006$]), and 16 negatives still lead (-0.009 [-0.017, -0.001]).

Table 20: Time split: filtered MRR over all 200,000 entities on held-out subjects (learned operators: three seeds). Recorded: facts of the snapshot; later answer: a fact the snapshot lacks for a query it answers; new query: a fact for a query it does not answer; new subject type: new-query facts whose subject is of a type never recorded for the relation. Verified: checked items (a random sample, plus every new-query item of the new subject type) whose fact was absent from the subject's Wikidata page at the end of July 2019. Oracle: the decoder given the true relation; matching: predicate matching; Cls_{16}: classification with 16 sampled negatives; Pred+z: four heads.

		no learned parameters				learned operator			
facts	n	oracle	matching	prototype	offset	Pred	Cls	Cls_{16}	Pred+z
recorded	20,000	0.972	0.972	0.141	0.183	0.707	0.694	**0.750**	0.611
later answer	10,673	0.239	0.239	0.066	0.069	0.180	0.178	**0.190**	0.155
new query	18,835	0.418	0.418	0.133	0.144	0.370	0.369	**0.379**	0.325
new subject type	723	0.345	0.345	0.057	0.092	0.226	0.236	**0.248**	0.208
later answer, verified	229	0.207	0.213	0.063	0.066	0.137	0.144	**0.146**	0.117
new query, verified	825	0.317	0.317	0.099	0.129	0.268	0.267	**0.282**	0.241
new subject type, verified	492	0.276	0.276	0.062	0.100	0.215	0.215	**0.234**	0.195

Table 21: Preregistered tests of the time split: MRR differences with 95% intervals (paired bootstrap over subjects, 1,000 resamples; H2: train and test subjects resampled independently). H2 is the difference in differences, Pred − Cls on later answers of training subjects minus the same on held-out subjects. H1 and H2 predict a positive difference, H4 a difference within 0.02; H3 has no registered direction. Oracle: the decoder given the true relation. New type: new subject type.

test	full sets	holds	verified subset	holds
H1: Pred − Cls, new query	$+0.001$ [-0.001, $+0.002$]	no	$+0.001$ [-0.006, $+0.007$]	no
H2: difference in differences	$+0.001$ [-0.002, $+0.003$]	no	$+0.014$ [-0.002, $+0.031$]	no
H3: matching − oracle, new type	-0.000 [-0.001, $+0.000$]	–	$+0.000$ [-0.000, $+0.000$]	–
H3: Pred − oracle, new type	-0.119 [-0.144, -0.095]	–	-0.061 [-0.087, -0.037]	–
H3: Cls − oracle, new type	-0.109 [-0.134, -0.085]	–	-0.061 [-0.087, -0.036]	–
H4: matching − oracle, new query	-0.000 [-0.000, $+0.000$]	yes	$+0.000$ [-0.000, $+0.000$]	yes
Pred − Cls_{16} (not registered)	-0.009 [-0.011, -0.006]	–	-0.015 [-0.025, -0.004]	–

E PREREGISTRATION, OUTCOMES AND DEVIATIONS

Every experiment had a protocol, written before its results, with explicit pass criteria (gates). When a design was changed after a result, the new design got its own protocol and is reported as a separate experiment, never as a re-analysis. Table 22 counts the gates by the finding they bear on, and each failed gate narrowed the corresponding claim in the paper. We do not correct intervals for the number of tests. Each claim in the main text rests on a registered gate whose outcome is reported whether it held or failed, and the gates are counted in full below, so a reader can weigh each claim against the number of tests behind it. The deviations below changed reported numbers, and Tables 23 and 24 at the end list every gate with its criterion and result.

Deviations that affect reported numbers.

Table 22: Preregistered gates by finding. Mixed: the gate holds for one model, split or chain set and fails for another. The time split has its own tests (Table 21), counted here under the first and third findings. Its H3 has no registered direction, so it counts as a gate of both findings without a verdict.

Finding	gates	met	mixed	not met	claim as stated in the paper
First: the predicted state behaves as a world model state	19	8	4	6	knowledge following for graphs and described systems, reuse with grounding
Second: state geometry decides whether to learn	13	6	0	7	role consistency predicts transfer, and learning is needed without it
Third: latent prediction against classification	13	2	2	8	latent prediction pays where classification sees few of many candidates

- **Graph conditions.** Before any model result, relation transfer (test sets A2, A3) was moved to the condition with all facts kept: under deletion the gold is a neighbour of the query for only 0.2–7% of items, so operators that attend over neighbours cannot answer by construction. Deletion is kept for unobserved correspondences.
- **Seeds.** Seed 0 of the learned graph models overstated the difference between Pred and Cls; all reported numbers are means of three seeds. Graph edits were registered on seed 0, and seeds 1 and 2 were added after review (Table 4 reports the mean, and every verdict is the same in each seed). The registered intervals resample clusters of seed-averaged differences; after the analysis we also resampled seeds and clusters together (Table 16), which leaves the gap to one negative above zero and makes the comparisons with more negatives on split A inconclusive.
- **Stopped target.** In the first runs of the learned centred offset, the Pred loss also trained the target projection. All reported Pred numbers for this operator use an EMA target; the earlier numbers, including a layer-7 gain that did not hold with the stopped target, are superseded. The registered learning gates on renamed systems (Table 23) used an earlier operator, a rank 16 adapter on the centred offset regressed onto the frozen states (top-1 0.30, follow 0.60), and Table 14 reports the learned centred offset.
- **MQuAKE edits.** In a first attempt the edited fact carried the weight of one fact among hundreds and was not learned; the edit was reweighted, relearning stops once it is fit, and the sham takes as many steps. The sham rate on chains that were correct before the edit is a post-hoc analysis, reported next to the registered rate.
- **Metrics changed before any full run.** Invented systems use role-level metrics for the open-set test, and the latent-variable test uses R-precision instead of the share of golds among the top 10, which was at ceiling in a smoke test.
- **Ties and function vectors.** Ties with a non-gold candidate count as errors throughout. The function-vector layer was chosen on the method's own tasks; on Pythia the method does not validate, so no SCAR number is reported.
- **Time split.** The registered validity check found 20 of 50 facts missing from the snapshot already on Wikidata in July 2019. Before any model was trained, the sets were relabelled from facts added later to facts missing from the snapshot, and a verified subset was added for the claim that facts were added later. The interval of the difference in differences (H2), which the first analysis script did not compute, was added after the first run.
- **Role consistency across models.** The first Gemma encoding stored float16 states, which overflowed at one layer; Gemma and Qwen3-8B were re-encoded in float32 before the analysis was used. The registered interval resamples the 49 points, which are seven layers of each of seven models. After the decision we also resampled models, so that the layers of a model enter together, and report that interval in the main text ([0.17, 0.67]).

Full log of preregistered gates.

Table 23: Preregistered gates of the experiments in the controlled world, on Wikidata graphs and on renamed systems.

What is tested	Criterion	Result	Verdict
Controlled world			

continued on the next page

Table 23 continued

What is tested	Criterion	Result	Verdict
Knowledge states encode roles	a role probe trained on training domains works on held-out domains	0.60–0.91 (chance 0.20)	pass
Operator that reads the query as a single vector	held-out transfer above the best parameter-free baseline	Pred 0.64, Cls 0.52 vs. 0.67	fail
Operators that read the query's components (component attention, transition attention)	same criterion	0.92–0.97 vs. 0.67	pass
Edits of the knowledge	follows edit ≥ 0.70, unrelated predictions stable ≥ 0.80, sham ≤ 0.10	component attention Pred: 0.72 / 0.95–0.98 / 0.00; transition attention Pred: sham 0.11	component attention pass, transition attention fail
Reuse, output restricted to known states (transition attention)	Pred ahead of Cls after two latent steps	gap 0.08 vs. 0.07	fail
Reuse, unconstrained output (transition attention with residual)	Pred $-$ Cls > 0 after two latent steps	$+0.50$, $+0.37$ (two worlds)	pass
Choosing between two readings of a role	correct reading on held-out systems	0.84 / 0.88, 0.82 / 0.85	pass
Wikidata graphs (WikiTopics, MQuAKE)			
Unseen relations, facts known (test set A2)	learned operator $\geq$ best parameter-free baseline $+0.05$ on both splits	Pred $+0.31$, $+0.25$	pass
Same role through another property (test set A3)	same criterion	$+0.08$ (split A), -0.07 (split B)	fail
Deleted facts: more than graph proximity	on items still joined by a path of ≤ 2 edges, beat personalised PageRank	predicate matching 0.18 / 0.21 vs. 0.003 / 0.007	pass
Deleted facts: learned operator (transition attention with support fit)	within 0.02 of predicate matching	Pred 0.20, Cls 0.21 vs. 0.29	fail
Counterfactual edits (MQuAKE)	follows edit ≥ 0.70, unrelated chains stable ≥ 0.80, sham ≤ 0.10	seed 0: 0.75 / 0.97 / 0.14, three seeds: 0.78 / 0.97 / 0.17	pass / pass / fail
Renamed systems (language-model states)			
Centred offset follows edits	follows ≥ 0.5, sham ≤ 0.1	Qwen 0.65 / 0.09; Pythia 0.37 / 0.03	Qwen pass, Pythia fail
Learning improves on the centred offset	learned Pred $\geq$ parameter-free $+0.05$	-0.04	fail
Learned operator follows edits	follows ≥ 0.5, sham ≤ 0.1	0.60 / 0.09	pass
Learned operator robust to reordering	prediction kept ≥ 0.8 under both reorderings	0.58 / 0.80	fail
Early layer (7): manipulation check	parameter-free accuracy drops by ≥ 0.10 from layer 21	0.34 $\rightarrow$ 0.13	pass
Early layer (7): learning recovers	learned Pred (stopped target) $\geq$ parameter-free $+0.10$	$+0.00$	fail
Random linear map of layer 21: manipulation check	parameter-free accuracy drops by ≥ 0.10	0.34 $\rightarrow$ 0.40	fail

Table 24: Preregistered gates of the experiments on invented systems, on reuse, on the conditions under which the JEPA objective helps, on the time split, and on the objective and role-consistency tests of Appendix D (three seeds unless noted).

What is tested	Criterion	Result	Verdict
Invented systems (Qwen, Pythia)			

continued on the next page

Table 24 continued

What is tested	Criterion	Result	Verdict
Transfer to invented systems	parameter-free $\geq$ chance $+0.15$	0.88, 0.41 (chance 0.11)	pass
Transfer needs the description	with names only, $\leq$ chance $+0.05$	0.10, 0.11	pass
Relation rather than attribute similarity	in cross-mapped systems, where an attribute points to a distractor: centred offset $\geq$ prototype $+0.10$	$+0.02$, $+0.02$ (both succeed)	fail
Follows edits of the description	follows ≥ 0.5, sham ≤ 0.1	0.94 / 0.04, 0.70 / 0.04	pass
Graded validity	full holder $>$ partial holder $>$ others in ≥ 0.6 of items	0.79, 0.23	Qwen pass, Pythia fail
Learning helps	learned Pred (stopped target) $\geq$ parameter-free $+0.05$	0.98, 0.66	pass
Reuse of the predicted state			
Two steps in invented systems	Pred $-$ Cls ≥ 0.10 after the latent rollout	-0.04, -0.11	fail
Three hops on MQuAKE, trained on its own rollouts	accuracy lost against teacher reset (drift) ≤ 0.10	0.27	fail
When the JEPA objective helps			
JEPA objective as the entity encoder	encoder fits the facts (hits@1 ≥ 0.99) before comparing role geometry	0.65–0.79 (three variants)	fail (comparison void)
Verifying an observation by a threshold	Pred $>$ Cls in the area under the ROC curve when one threshold on cos(prediction, observed entity) separates true from false answers	Cls $\geq$ Pred or both at ceiling	fail
Latent variable, graph states	Pred+z $>$ Pred on test set A2 and Pred+z $\geq$ Cls -0.01	-0.09, -0.10	fail
Latent variable, MQuAKE latent reuse	Pred+z $\geq$ Pred $+0.10$ after three latent hops (predicted to fail)	$+0.03$	fail
Latent variable, unique answers (control)	$\lvert$Pred+z $-$ Pred$\rvert \leq 0.02$	$+0.00$, $+0.01$	pass
One negative per query	Pred $\geq \text{Cls}_1$ (interval not below 0)	graph $+0.13$, $+0.12$; MQuAKE -0.03; invented -0.00, -0.10	pass on graphs only
No candidate set in training	Pred $\geq \text{Cls}_\text{b}$ (interval not below 0)	graph -0.02, $+0.14$; MQuAKE -0.02; invented -0.01, $+0.02$	pass on graph B and Pythia
Why Pred ranks all golds higher (follow-up)	if Pred $-$ $\text{Cls}_\text{avg} \leq 0.03$ in R-precision, the gap is due to how Cls aggregates golds	$+0.00$, $+0.03$	due to aggregation
Time split of Wikidata			
Four registered hypotheses (H1 to H4)	see Table 21	H1, H2 not met; H3 reported; H4 met	1 of 3
Objectives on a 200,000-entity graph; role consistency across language models			
Held-out relations favour Pred	held-out relations: Pred $>$ Cls_1 and Pred $>$ Cls_{16} (intervals above 0)	$+0.006$ [-0.001, $+0.012$]; -0.035	fail
Relation novelty moderates Pred $-$ Cls_n	(held-out $-$ seen) > 0 for $n = 1$ and 16	-0.064; $+0.010$	fail
No Pred advantage on seen relations	Pred $-$ $\text{Cls}_{16} \leq 0.01$	-0.044	pass
Shrinking with n on held-out relations	Pred $-$ $\text{Cls}_1 \geq$ Pred $-$ $\text{Cls}_{16} \geq$ Pred $-$ Cls_{256}	$+0.006$, -0.035, -0.034	fail
κ predicts the centred offset	Spearman over 49 (model, layer) points > 0, interval above 0	0.45 [0.19, 0.65], models resampled [0.17, 0.67]	pass
Best-κ layer per model	top-1 at the highest-κ layer $\geq$ the model's median	5 of 7 models	fail