FACTGRASP: GRAPH-AWARE RELATIONAL ACTIVATION SHAPING FOR FACTUAL KNOWLEDGE DISCOVERY, RECALL, AND TRANSFER IN LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Language Models (LLMs) implicitly encode factual knowledge in their hidden representations, which are entangled across layers and other learned information, making them difficult to discover and reinforce. Existing solutions, such as retrieval augmentation, external probing, and knowledge editing, either leave the hidden states unchanged, provide a mechanism for diagnosis without reinforcing knowledge recall or modify parameters in a way that may cause interference. We propose Graph-Aware Activation ShaPing (FactGRASP) as a method to linearly transform activations of LLM's hidden states with explicit Knowledge Graph (KG) triples. FactGRASP learns a linear map of subject-relation to object representations using a KG-contrastive objective with and without relation-aware gating. The two major objectives are (i) to provide a diagnostic tool to identify the types of factual associations that are linearly detectable, and (ii) to reshape activations to improve model representations in knowledge-rich settings. We evaluate the approach on KG completion across 19 LLMs, where the object representation obtained by simply composing the subject and relation representations leads to 0.064 MRR. With the learned linear map this increases to 0.405 and with relation-aware gating to 0.417 MRR. We further show that a trained linear map can be reused across models with an MRR comparability of up to 95.9%. Additional experiments examine when relational predictions become accessible during training and whether factual associations are consistently difficult across models. The results demonstrate that graph-aware activation shaping is a lightweight mechanism to make factual associations more accessible and available in language models.

1 INTRODUCTION

Large Language Models (LLMs) learn to encode real-world facts through exposure to natural language during training. Such facts are entangled across layers and learned representations and complex analyses are required to explicate and interpret them. Mechanistic interpretability offers a lens to many complex phenomena that arise during LLM training, including factual recall and construction in training or inference. Research on factual knowledge in LLMs focuses on interpreting where facts are stored and how they are constructed when real-world questions are posed to the model (Wang et al., 2025). Representations in attention heads and Multi-Layer Perceptrons (MLPs) can also be used to perform relation classification guided by linear probes (Popovič & Färber, 2026) and relation-specific neurons have been found to drive generation involving specific relations (Liu et al., 2025). Nevertheless, mechanistic interpretability approaches, in most cases, produce diagnostic tools and do not reinforce factual knowledge or make it more explicit for further use.

Inspired by previous work on knowledge recall and interpretability (Hernandez et al., 2024; Merullo et al., 2025), we propose a method to identify factual associations in transformer language models' hidden states using Knowledge Graph (KG) triples. We introduce Graph-Aware Activation ShaPing (FactGRASP), a method to make factual associations modeled by KG triples in the form of subject, relation, object (s, r, o) more accessible using a linear transformation of an LLM's hidden states. The transformation enables the reconstruction of the object o by simple vector composition of the representations of subject s and relation r. An even stronger reinforcement of factual knowledge is achieved with the proposed gating mechanism on a whole Mixture-of-Transformations (MoT), that is, relation-specific linear maps.

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.

Consider the fact that *Paris* is in *France*. A language model may answer a question about it, but its answer does not directly reveal how the subject, relation, and object are represented internally. FactGRASP evaluates whether the learned linear transformation improves the capacity of the model's representations to identify *France* from *Paris* and the relation *country*. Furthermore, we investigate if such a transformation learned from one model can be reused with another model after accounting for differences in their representations. Thereby, FactGRASP studies factual associations independently of whether the language model explicitly generates the corresponding answer.

In more explicit terms, given a KG triple (s, r, o), we encode the subject and relation with a frozen language model and compose their representations as $x = h_s + h_r$. A learned linear map transforms this composed vector representation toward the object representation h_o, and the resulting vector is evaluated by ranking candidate objects. We compare a general linear transformation with a relation-specific version of eight relations in an MoT version with gating. In all these experiments the weights of the underlying model are frozen. Thus, the learned transformation operates as a relational linear map rather than a modification of the language model.

Across 19 frozen language models, a contrastive training objective of the linear map building on cosine and InfoNCE (Oord et al., 2018) loss provides an MRR of 0.405 that is increased to 0.417 with relation-aware MoT with gating, compared to 0.064 before the transformation. This shows that the linear map achieves a strong improvement on factual representation, with even more increase by the relation-aware method. We further investigate whether the learned linear map is specific to a model's representation space. A transfer of these learned linear maps across models is investigated, achieving up to 95.9% comparability of MRR across models.

Our main contributions are:

1. We model the compositional structure of (s, r, o) triples directly in representation space, using a lightweight linear transformation on representations coming from an LLM to identify and rank factual objects without modifying the underlying language model.
2. We compare the linear map with the relation-aware MoT with gating as well as the cosine loss and its combination with contrastive InfoNCE loss across 19 LLMs.
3. We show that the learned linear maps can be reused in other models after alignment, keeping an MRR performance across models.
4. We further examine when relational prediction becomes accessible during training and whether models share difficulties on individual factual associations.

2 Related work

Closely related to the proposed approach are lines of research related to linear probes, knowledge transfer, and activation intervention. Since a full coverage of all of these topics goes beyond the scope of a single conference paper, we provide a brief account to better position our approach.

Linear factual structure and probing. A closely related line of work studies whether factual and relational knowledge in transformer language models exhibits linear structure. Hernandez et al. (2024) investigate how transformer language models retrieve and use relational knowledge and show that, for a subset of relations, the subject-to-object computation can be approximated by a relation-specific affine transformation. In their formulation, the relation is encoded in the transformation itself, whereas we explicitly represent the relation as a contextual embedding and learn a transformation over the composed subject-relation representation. Thus, rather than learning a separate subject-to-object operator for each relation, we model the compositional structure of knowledge graph triples (s, r, o) directly in representation space. This formulation is closer to knowledge graph embedding approaches, such as TransE, where relations are expressed through vector arithmetic.

Other studies have examined relational information in intermediate representations. Popovič & Färber (2026) identify latent representations in LLMs that support relation classification using linear probes, while Geva et al. (2023) study how models aggregate subject and relation information to predict attributes. Liu et al. (2025) further identify and ablate relation-specific neurons associated with generation involving particular relations. Our setting differs in that we use these representations to perform KG-style object prediction through a learned vector-to-vector transformation, rather than only classifying relations or identifying individual components associated with them.

Merullo et al. (2025) further study linear factual representations and relate the strength of such representations to pretraining frequency. Our focus is complementary in that rather than characterizing when a particular factual relation becomes linearly represented, we learn a linear map across many relations and models and investigate whether that map can be reused after cross-model alignment.

Cross-model representation alignment and transfer. A second relevant line of work asks whether representations learned by different neural networks can be aligned or transferred. Model stitching studies whether a learned transformation can connect representations from different networks while preserving downstream performance (Lenc & Vedaldi, 2015; Bansal et al., 2021). Representation-similarity and embedding-translation studies similarly investigate relationships between representation spaces across models (Huh et al., 2024; Jha et al., 2026). In contrast, we investigate whether a fixed, supervised relational transformation of one language model's hidden representations can be reused in another model.

Activation intervention and representation engineering. Our approach goes beyond probing and shapes activations. Activation interventions and causal representation analyses study whether manipulating a representation affects model outputs (Li et al., 2023; Geiger et al., 2024), while concept-erasure methods investigate the removal of information from representations (Ravfogel et al., 2020; Belrose et al., 2023). Representation engineering similarly uses directions in activation space to steer model behavior (Zou et al., 2023). Our intervention setting differs in that the manipulated direction is derived from the supervised relational linear map, rather than being an independently identified concept direction. Importantly, our experiments distinguish between the ability of a shaping map to predict an object and the ability of its derived direction to improve the model's own output; predictive success alone does not establish that the underlying language model implements or uses the learned shaping map.

3 METHOD

After a general problem formulation, we introduce the utilized KGs, evaluated models, and the linear transformation methods. We continue to explain the objectives we utilized to train the linear map and the relation-gated MoT as well as the method for selecting the best layer from which to derive the hidden representations for each model. Finally, we explain how we transfer the learned linear map across models and the evaluation method.

3.1 PROBLEM FORMULATION

We formulate factual association as a KG triple and the linear transformation as a KG completion problem. Each KG triple is represented as (s, r, o), where s is a subject, r is a relation, and o is the corresponding object, for example, $(\text{Paris}, \text{country}, \text{France})$. Given a subject-relation pair (s, r), our goal is to construct a representation that can rank the correct object among a set of candidate objects. For each triple, the subject, relation, and object are encoded separately by a frozen language model. Let $h_s, h_r, h_o \in \mathbb{R}^d$ denote their representations at a methodically selected hidden layer. No language-model parameters are updated during FactGRASP training, but only transformed.

3.2 DATASET

The KG triples are taken from Wikidata and ConceptNet (Speer et al., 2017) for comparison and detailed in Table 1. We use a subset of Wikidata5M (Wang et al., 2021) containing 42,422 triples over 191 relations, split into 33,937 training, 4,242 validation, and 4,243 test triples. In total, 2,766 unique object strings are contained to evaluate KG completion. We additionally use a subset of ConceptNet to examine whether the observed behavior extends to commonsense relations. Results on ConceptNet comparisons are reported in the appendices.

3.3 MODELS

The primary experiments evaluate 19 models from nine model families, including Gemma, OLMo, Llama, and Qwen, spanning approximately 0.15B to 12B parameters as listed in Table 2. For decoder models, the representation is obtained from the final non-padding token, while the ModernBERT experiments use mask-aware mean pooling. Intermediate decoder representations are passed

Table 1: Datasets split into training, validation, and test triples, number of relations, and unique object strings.

Graph	Train	Val	Test	Relations	Objects
Wikidata subset	33,937	4,242	4,243	191	2,766
ConceptNet subset	29,081	3,635	3,635	9	3,177

through the model's final normalization module before comparison. For models with a dimensionality exceeding 768, we project the representations to a common 768-dimensional space using PCA fitted only on the training data. Thus, the input to FactGRASP consists entirely of representations extracted from a frozen language model.

3.4 Relational Composition and Activation Shaping Transformations

Our formulation is inspired by the additive composition used in knowledge graph embedding methods, such as TransE (Bordes et al., 2013), where subject and relation representations $h_s, h_r \in \mathbb{R}^d$ are combined to approximate the object representation $h_o \in \mathbb{R}^d$. Rather than assuming that the LLM representation space already satisfies this additive structure, FactGRASP learns a linear transformation that maps the subject-relation composition toward the corresponding object representation. To this end, we first form the composed subject-relation vector, hereafter referred to as the composed vector, as shown in Equation 1.

$$x = h_s + h_r. \tag{1}$$

A linear activation shaping transformation $W \in \mathbb{R}^{d\times d}$ is learned to improve the prediction of the corresponding object representation as in Equation 2,

$$\hat{h}_o = Wx, \tag{2}$$

where $\hat{h}_o \in \mathbb{R}^d$ is the estimate of the ground-truth object vector $h_o \in \mathbb{R}^d$. The relation is already a part of the input through h_r, allowing the same transformation to model different relations. We denominate this single W learned for all relations generically a linear map. This setup provides a simple test of whether relational structure is linearly accessible after a shared transformation. We further consider whether relation-dependent transformations provide additional capacity. We, therefore, introduce MoT as a mixture of eight linear transformations or experts $\{W_i\}_{i=1}^8$ as shown in Equation 3. We use eight transformations as this is the number of unique relations we used for these experiments.

$$\hat{h}_o = \sum_{i=1}^{8} g_i W_i x, \tag{3}$$

The relation representation h_r determines the mixture weights through Equation 4,

$$g(h_r) = \mathrm{softmax}(Gh_r + b), \tag{4}$$

where $b \in \mathbb{R}^8$ is the bias vector, $G \in \mathbb{R}^{8\times d}$ is a linear map followed by a softmax layer that transforms $h_r \in \mathbb{R}^d$ to $g(h_r) \in \mathbb{R}^8 = [g_1, g_2, ...g_8]^T$, such that the individual gating entries g_is are non-negative and sum to one, i.e., $\sum_{i=1}^8 g_i = 1$. Since g_is are derived from the relation representation h_r, these are used to control how much of each $W_i x$ will contribute to $\hat{h}_o$ in Equation 3 as shown in Fig.1. Thus, we compute the estimated $\hat{h}_o$ via the relation-gated MoT in Equation 3.

In the relation-gated MoT, the gate receives the weights through the relation representation and not a discrete relation identifier, so the formulation remains defined for relation representations that were not explicitly assigned an expert during training. In other words, the approach can generalize to unseen relations. More specifically, we present comparisons through:

1. a raw estimate of $\hat{h}_o = h_s + h_r$, with an objective to determine if there exists a representation in the LLM's hidden representations such that the sum of $\mathbf{h}_s$ and $\mathbf{h}_r$ yields a close estimate of the object embedding $\mathbf{h}_o$,
2. an estimate based on a linear map defined as $\hat{h}_o = Wx$, and
3. a relation-gated mixture of transformations (MoT) based estimate $\hat{h}_o = \sum_{i=1}^8 g_i W_i x$ where eight linear activation shaping transformations $\{W_i\}_{i=1}^8$ are learned and gated via the weights learned through the relation representation h_r.

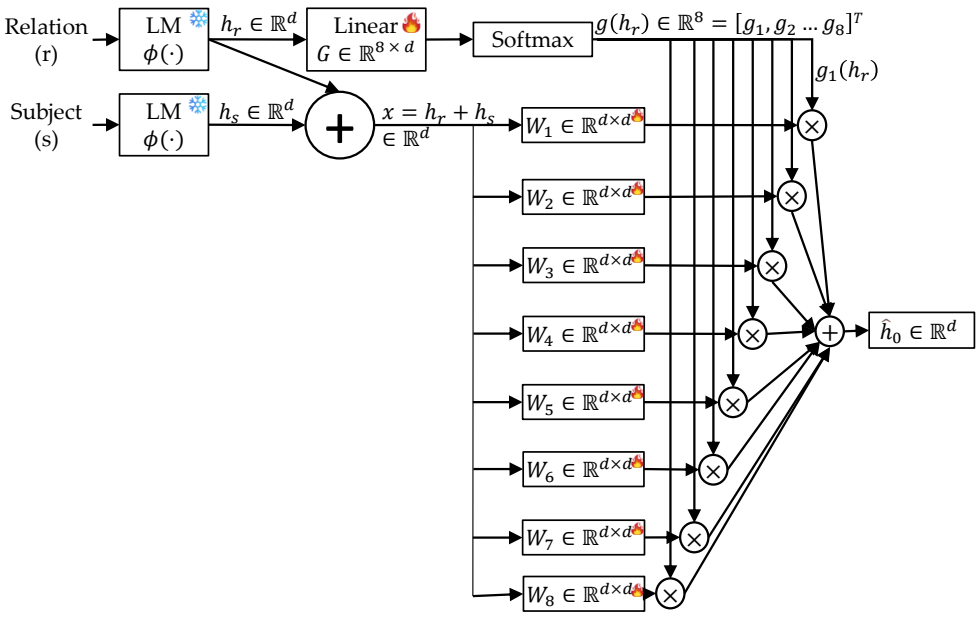


Figure 1: A relation-gated mixture of transformations.

All activation shaping transformations were learned using AdamW with learning rate 10^{-3}, weight decay 10^{-4}, batches of $B = 512$, and a maximum of 60 epochs. The learning rate is halved after four validation plateaus, with early stopping after twelve.

3.5 Training Objectives

A prediction close to the gold object in cosine similarity need not rank that object above competing candidates. We, therefore, compare a pointwise cosine objective with a candidate-aware contrastive objective. The pointwise objective minimizes the cosine distance between the transformed representation object ($\hat{h}_o$) and the ground-truth object (h_o) representations as in Equation 5.

$$\mathcal{L}_{\cos} = \frac{1}{B} \sum_{i=1}^{B} \Big(1 - \cos(\hat{h}_{o_i}, h_{o_i})\Big) \tag{5}$$

For the contrastive objective, we augment the cosine objective with an in-batch InfoNCE (Oord et al., 2018) loss as shown in Equation 6.

$$\mathcal{L}_{\cos} + \mathcal{L}_{\text{InfoNCE}} = \mathcal{L}_{\cos} - \frac{1}{B} \sum_{i=1}^{B} \left[\log \frac{\exp\big(\cos(\hat{h}_{o_i},\, h_{o_i}) \,/\, \tau\big)}{\sum_{j=1}^{B} \exp\big(\cos(\hat{h}_{o_i},\, h_{o_j}) \,/\, \tau\big)} \right] + \lambda. \tag{6}$$

In Equation 6, $\tau = 0.05$ is a temperature hyperparameter, $B = 512$ is the batch size, and $\lambda > 0$ is a weighting hyperparameter. $\{h_{o_j}\}_{j=1}^{B}$ is the set comprising the positive and $B - 1$ negative samples. After initial experiments (see Appendix A), the default training objective is selected as cosine distance plus in-batch InfoNCE, with equal weighting of the two terms. The relation-gated MoT additionally uses a gate-balancing penalty of $0.01 \sum_{e=1}^{8} \bar{g}_e \log(8\bar{g}_e)$, where $\bar{g}_e$ is the mean gate weight in the batch, to discourage collapse onto a single expert.

3.6 Layer Selection

LLMs provide representations at multiple hidden layers, and the layer producing the most useful relational representation need not be the last layer (Skean et al., 2025; Batra et al., 2026). For

each model, we sweep the candidate hidden layers using a linear map with combined cosine and InfoNCE loss and select the layer with the highest validation MRR, as detailed in Appendix B. For a subset of models, we test the four configurations of linear map and MoT with respectively cosine and combined cosine and InfoNCE loss. Detailed experiments for layer selection with Wikidata and also ConceptNet for comparison are presented in Appendix B.

3.7 Knowledge Transfer

Reference model A supplies the activation shaping transformation W_A and candidate object vectors. Input model B supplies the composed vector. For example, a composed vector encoded by Qwen may be aligned with Llama, passed through Llama's activation shaping function, and compared with Llama's object vectors as shown in Fig. 2. This choice of object vectors is what we call the candidate space and the transformed representation of model A is called a reference space. A ridge-fitted alignment maps composed vectors into the reference space as shown in Equation 7.

$$S_{B\to A} = \arg\min_S \sum_i \|Sx_i^B - x_i^A\|_2^2 + \lambda\|S\|_F^2. \tag{7}$$

The transferred composed vector and its linear map is $W_A S_{B\to A} x^B$, ranked against A's candidates. Each pair of models needs its own alignment. We use the subject and relation of the entire training dataset for the experiments. A pairing of models for the purpose of this experiment resulted in 342 individual combinations of model A and B given that we experimented with 19 models. Further detailed experiments on the impact of data size and model pairing are presented in Appendix D.

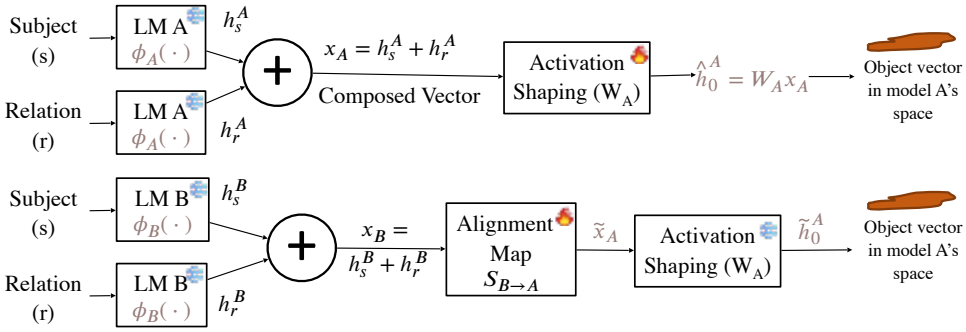


Figure 2: Transfer of learned linear maps across models by a pair-wise comparison. The frozen model embeds the subject and relation separately; a learned map transforms their sum to rank object vectors. For transfer, model B supplies the composed vector and a fitted alignment $S_{B\to A}$ maps it into the reference model A's space. Model A supplies both the fixed activation shaping transformation W_A and the candidate object vectors.

3.8 Evaluation

We utilize Mean Reciprocal Rank (MRR) and Hits@k, used as standard metrics in information retrieval, to evaluate the KG completion performance. Since a subject and relation compositions may have more than one correct object, we evaluate one answer and remove the other known correct answers from its competitors, using facts in the training, validation, and test splits (Bordes et al., 2013; Sun et al., 2020). For example, if two people are known cast members of a film, ranking one above the other should not count as an error.

For Hits@k we use k=1 and k=10. We compare the composed vector of the original hidden representation with the linearly mapped representation and the relation-gated MoT, using the same gold-truth object vectors. We average obtained scores over three random initializations with seeds for W.

4 Results

After the linear transformation and knowledge transfer results, we introduce analyses of accessibility and difficulty of factual associations per model.

4.1 Linear Map and Mixture-of-Transformations

At the selected layers across the 19 models, a linear map reaches a mean over 19 models MRR of 0.405, compared with 0.064 for the composed vector without any transformation. The relation-gated MoT reaches 0.417 as shown in Table 2. The same pattern holds for Hits@k with mean Hits@1 rising from 0.035 to 0.316, and Hits@10 from 0.117 to 0.614. The largest MRR gain can be observed for Llama-3.1-8B, from 0.085 to 0.503 with MoT. We would like to emphasize that the models remain frozen and these improvements come from training a linear map. Across models, the MoT adds 0.353 MRR relative to the composed vector of original representations, where the linear map alone captures most of this gain. An ablation study for the linear map is shown in Appendix C.

As shown in previous research (Skean et al., 2025; Batra et al., 2026; Hernandez et al., 2024), intermediate layers might hold richer representations than the last layer. Selecting the layer with the best validation MRR, rather than always using the last layer, improves the linear map's MRR by 0.016 on average (more details on this comparison can be found in Appendix B). These results confirm that earlier layers rather than the final ones provide a representation that performs best on our KG completion task.

Table 2: Knowledge graph completion results on the test set. Object representation refers to the result of the subject-relation composed vector from untransformed hidden states. Layer denotes the selected one over the total number of layers. Bold marks the best performance over evaluation metrics. The activation shaping transformations are initialized using three different seeds and the mean of MRR and Hits over these three different trained transformations are used.

		Object Representation			FactGRASP			
					Linear Map	MoT		
Model	Layer	MRR	Hits@1	Hits@10	MRR	MRR	Hits@1	Hits@10
Gemma-4-E2B	27/35	0.056	0.029	0.110	0.410	**0.417**	**0.317**	**0.620**
Gemma-4-E4B	33/42	0.056	0.030	0.103	0.447	**0.453**	**0.348**	**0.659**
Gemma-4-E4B-it	5/42	0.037	0.023	0.058	0.273	**0.281**	**0.199**	**0.444**
Gemma-4-12B	39/48	0.063	0.037	0.111	0.463	**0.473**	**0.368**	**0.674**
Granite-4.2-8B	40/40	0.033	0.014	0.070	0.295	**0.319**	**0.228**	**0.504**
Llama-3.1-8B	25/32	0.085	0.047	0.155	0.497	**0.503**	**0.394**	**0.716**
Llama-3.1-8B-Instruct	23/32	0.088	0.051	0.159	0.493	**0.499**	**0.390**	**0.711**
Llama-3.2-1B	11/16	0.067	0.038	0.119	0.427	**0.435**	**0.330**	**0.642**
Llama-3.2-3B	19/28	0.072	0.041	0.129	0.457	**0.469**	**0.363**	**0.680**
Ministral-3-8B-Instruct	27/34	0.071	0.039	0.134	0.439	**0.449**	**0.342**	**0.660**
ModernBERT-base	3/22	0.045	0.029	0.072	0.269	**0.300**	**0.230**	**0.435**
OLMo-3-7B	27/32	0.067	0.038	0.119	0.423	**0.435**	**0.331**	**0.636**
OLMo-3-7B-Instruct	23/32	0.061	0.034	0.107	0.371	**0.389**	**0.291**	**0.584**
Qwen3.5-0.8B	21/24	0.065	0.035	0.124	0.358	**0.366**	**0.272**	**0.560**
Qwen3.5-2B	21/24	0.072	0.036	0.138	0.387	**0.400**	**0.301**	**0.595**
Qwen3.5-4B	26/32	0.076	0.042	0.140	0.428	**0.434**	**0.330**	**0.636**
Qwen3.5-4B-Base	28/32	0.075	0.041	0.138	0.436	**0.440**	**0.332**	**0.651**
Qwen3.5-9B	28/32	0.078	0.041	0.150	0.470	**0.476**	**0.370**	**0.681**
SmolLM3-3B	35/36	0.050	0.026	0.096	0.345	**0.376**	**0.276**	**0.577**
Mean over 19 models		0.064	0.035	0.117	0.405	**0.417**	**0.316**	**0.614**

As shown in Table 3, we utilize four configurations: linear map with cosine loss, relation-gated MoT with cosine loss, linear map with cos+InfoNCE loss, and relation-gated MoT with cos+InfoNCE loss. Under cosine-only training following Equation 5, the shared map scores 0.290 MRR and MoT 0.361. The introduction of contrastive InfoNCE improves the performance of both approaches, while at the same time shrinking their difference (see Table 3).

Linear maps and MoT reach an MRR of 0.4–0.5, showing that training these maps learns useful knowledge-rich representations. The pretraining of these maps alone with untrained encoder of BERT-base is responsible for an increase in performance of 0.355 versus 0.226 (see Table 8, Appendix C). Because the map learns from correct answers, it can pick up some patterns even from meaningless vectors of untrained LM model. Thus, these scores show how much gain in MRR comes from the trained linear map alone. When evaluating on completely unique triples where no

Table 3: Mean test performance across 19 models at the selected preferred layer with different training objectives.

Configuration	MRR	Hits@1	Hits@3	Hits@10
Linear map, cosine	0.290	0.210	0.313	0.447
MoT, cosine	0.361	0.277	0.388	0.526
Linear map, $\cos + \text{InfoNCE}$	0.405	0.302	0.446	0.608
MoT, $\cos + \text{InfoNCE}$	**0.417**	**0.316**	**0.458**	**0.614**

entity or relation is repeated across dataset splits, the contrastive MoT reaches 0.329, 0.252, and 0.313 for Qwen3.5-4B, SmolLM3-3B, and Gemma-4-E4B respectively, versus 0.305, 0.228, and 0.260 for matched cosine-trained MoTs (see Table 8, Appendix C).

4.2 Results of Knowledge Transfer

An activation shaping map trained on model A's representations can be reused on a model B's composed vectors without retraining, following the method described in Section 3.7. Calculating the MRR across 342 model pairs from all 19 models, the comparability of the MRR reaches up to 95.9% (Fig. 3), with a median absolute MRR change of -0.017 and an interquartile range $[-0.059, 0.004]$.

We carry out an 8-pair comparison, that is, three within a family, e.g. Qwen3.5-4B's map is used on Qwen3.5-9B's composed vectors, three across families, e.g. Llama-3.1-8B's map is used on Gemma-4-12B's composed vectors, and two where the LM A is the encoder BERT-base that is outside the main 19 LMs. The detailed results are presented in Appendix D. On every pair, reusing A's existing linear map through the alignment reaches MRR of 0.969 to 0.987 of the trained map's score on the same candidate pairs. Thus, we observe that transferring the fixed reference map from LM A through alignment performs nearly as well as training a new map for LM B on the same reference candidate vectors. Deliberately mismatching the paired composed subject-relation vectors reduces median transfer efficiency to 0.007.

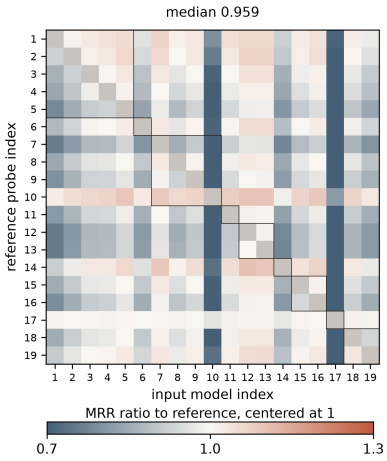


Figure 3: MRR relative to the reference activation shaping map's native score for every ordered model pair; rows refer to the candidate vectors of LM Model A, columns refer to the composed vectors of LM Model B. The grey diagonal omits same-model pairs. Model indices and family outlines are defined in Table 10 in Appendix D.

Fig. 3 shows the MRR for model pairs, where Columns 10 (Gemma-4-E4B-it) and 17 (ModernBERT-base) show weaker transfer into other models' reference spaces. For the other pairings the transfer is similarly effective as detailed also in Appendix D, including the impact of dataset size, that is, number of triples, on the knowledge transfer. The alignment remains effective with substantially fewer examples. For instance, randomly sampling 18,761 paired composed vectors from the original 33,937 retains a median MRR ratio of 95.8%, compared to 95.9% for the full set. In contrast, additionally removing some test pairs from the alignment data drops this ratio to 90.7%.

The size-matched random subset shows that this additional loss comes from which examples are removed, rather than the smaller training set alone.

Experiments show that the knowledge transfer is also effective beyond transformer architectures. We align the composed vectors coming from LLama's representations with the composed vectors coming from a Falcon-Mamba-7B, a non-transformer LLM. After alignment, the Llama activation shaping map reaches 0.433 MRR when applied to Mamba's composed vectors, compared with 0.427 MRR for Mamba's own native performance on its own vectors.

4.3 Accessibility changes through training and Fact-level agreement

We perform the KG completion task on 28 OLMo-2-1B pretraining checkpoints plus the final released model, to analyze how the performance of the activation shaping map evolves over the course of pretraining. At the earliest checkpoint (one billion tokens), the activation shaping map performs close to the baseline without any linear transformation. By the next available checkpoint (21 billion tokens), it has already reached 72% of the final released model's gain above that baseline. Fig. 4(a) shows this performance development on OLMo-2-1B, alongside the transfer from each OLMo checkpoint to the fixed final Llama model, since Llama did not release pretraining checkpoints. The two trajectories develop in parallel, with Llama showing a consistently higher MRR throughout. Fig. 4(b) shows that training changes KG completion performance, while different models partly agree on fact difficulty. Further details are presented in Appendix E, where we test whether the linear map outperforms prompting of the same models, which it does.

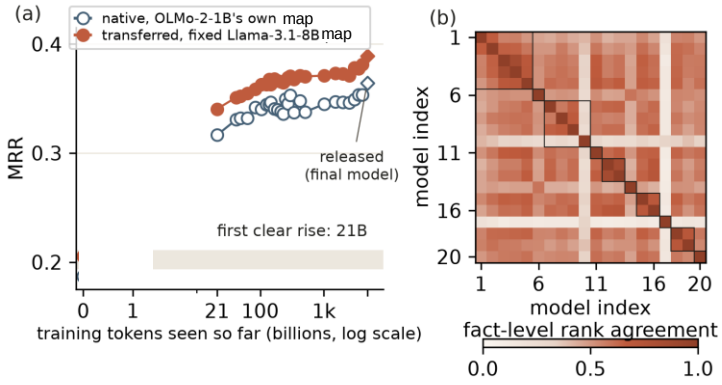


Figure 4: (a) Comparative MRR performance of the pretrained OLMo-2-1B model with a linear map trained on the model itself and with a linear map trained on the Llama-3.1-8B linear map for comparison. (b) Hits@1 scores between fact rankings within the same relations for 20 models (19 main models + Falcon-Mamba-7B). Model indices follow Table 10 in Appendix D.

In the 20-model fact-agreement analysis, comprising 19 main models and Falcon-Mamba, the median fact-ranking correlation is 0.549, overall ranging between 0.497–0.569 for different models (Fig. 4b), demonstrating shared task difficulty across models. Shaping maps of the same model initialized using different seeds agree much more strongly, at 0.977.

5 Conclusion

FactGRASP successfully shapes LLM activations to make factual associations more accessible and available. However, this accessibility of facts with the linear map is not comparable to targeted and reliable fact editing. We tested two main variants of FactGRASP: a linear map for all relations and a relation-gated MoT. The latter outperforms the former for all 19 evaluated models, however, the biggest performance gain compared to the baseline is achieved with the linear map. The baseline is the object representation derived by composing the subject and relation representation. Most interestingly the learned map can be used to transfer knowledge between LLMs as shown by a pair-wise knowledge transfer comparison.

AI USE STATEMENT

LLM assistance was partially used to revise the text of some sections of the paper, especially the appendices, and to create and modify visual aids in the form of figures. LLMs were further used to create and edit software code for the experiments and analyses. This usage was always strictly supervised and tested by human authors. Additionally, we identified and initially analyzed existing literature with LLMs, but the contents of cited papers were analyzed by human authors.

ETHICS STATEMENT

This work analyzes frozen public language models on public knowledge graph subsets derived from Wikidata and ConceptNet; it involves no human subjects and no personal data beyond what is contained in these two KGs. A linear map utilizing these graphs inadvertently also suffers from any biases or issues with factual reliability of the KGs.

REPRODUCIBILITY STATEMENT

Section 3 provides major details to reproduce these reported results. Furthermore, we will make all KG triple partitions and software code used for the experiments in this paper publicly available in its final version [1].

REFERENCES

Yamini Bansal, Preetum Nakkiran, and Boaz Barak. Revisiting model stitching to compare neural representations. In *Advances in Neural Information Processing Systems*, 2021.

Arnesh Batra, Arush Gumber, Aniket Khandelwal, Jashn Khemani, and Anubha Gupta. Uncovering the latent potential of deep intermediate representations. In *International Conference on Machine Learning*, 2026.

Nora Belrose, David Schneider-Joseph, Shauli Ravfogel, Ryan Cotterell, Edward Raff, and Stella Biderman. Leace: perfect linear concept erasure in closed form. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper_files/paper/2013/file/1cecc7a77928ca8133fa24680a88d2f9-Paper.pdf.

Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah D. Goodman. Finding alignments between interpretable causal variables and distributed neural representations. In *Proceedings of the Third Conference on Causal Learning and Reasoning (CLeaR)*, 2024.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. Dissecting recall of factual associations in auto-regressive language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12216–12235, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.751. URL https://aclanthology.org/2023.emnlp-main.751/.

Evan Hernandez, Arnab Sen Sharma, Tal Haklay, Kevin Meng, Martin Wattenberg, Jacob Andreas, Yonatan Belinkov, and David Bau. Linearity of relation decoding in transformer language models. In *International Conference on Learning Representations*, volume 2024, pp. 10504–10526, 2024.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. The platonic representation hypothesis. In *International Conference on Machine Learning*, 2024.

[1] Placeholder for a GitHub link.

Rishi Jha, Collin Zhang, Vitaly Shmatikov, and John Morris. Harnessing the universal geometry of embeddings. *Advances in Neural Information Processing Systems*, 38:45963–45987, 2026.

Karel Lenc and Andrea Vedaldi. Understanding image representations by measuring their equivariance and equivalence. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems 36*, 2023.

Yihong Liu, Runsheng Chen, Lea Hirlimann, Ahmad Dawar Hakimi, Mingyang Wang, Amir Hossein Kargaran, Sascha Rothe, François Yvon, and Hinrich Schuetze. On relation-specific neurons in large language models. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 992–1022, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.52. URL https://aclanthology.org/2025.emnlp-main.52/.

Jack Merullo, Noah Smith, Sarah Wiegreffe, and Yanai Elazar. On linear representations and pretraining data frequency in language models. In *International Conference on Learning Representations*, volume 2025, pp. 71963–71987, 2025.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR, abs/1807.03748*, 2018.

Nicholas Popovič and Michael Färber. Tracing relational knowledge recall in large language models. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 43490–43509, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.2160. URL https://aclanthology.org/2026.findings-acl.2160/.

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. Null it out: Guarding protected attributes by iterative nullspace projection. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7237–7256, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.647. URL https://aclanthology.org/2020.acl-main.647/.

Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. Layer by layer: uncovering hidden representations in language models. In *Proceedings of the 42nd International Conference on Machine Learning*, ICML'25. JMLR.org, 2025.

Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

Zhiqing Sun, Shikhar Vashishth, Soumya Sanyal, Partha Talukdar, and Yiming Yang. A re-evaluation of knowledge graph completion methods. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5516–5522, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.489. URL https://aclanthology.org/2020.acl-main.489/.

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. doi: 10.1162/tacl_a_00360. URL https://aclanthology.org/2021.tacl-1.11/.

Yiqun Wang, Chaoqun Wan, Sile Hu, Yonggang Zhang, Xiang Tian, Yaowu Chen, Xu Shen, and Jieping Ye. Tracing and dissecting how LLMs recall factual knowledge for real world questions. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar

(eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23246–23271, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1133. URL https://aclanthology.org/2025.acl-long.1133/.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to AI transparency. *CoRR abs/2310.01405*, 2023.

A Loss Function Experiments

We performed initial experiments on BERT-base for selecting a suitable loss function to optimize the linear map in order to retrieve the correct object from a set of candidates. Table 4 shows the results of two different loss functions and their combination. Training only on the cosine function reaches satisfactory results, but a big performance increase is reported when training the activation shaping layer on the InfoNCE loss (Oord et al., 2018), reaching a 34.4% increase in MRR. An additional 2.44% MRR increase is reported when combining both losses as described in Equation 6, which is the reason why it was ultimately selected as loss function for all other experiments.

Table 4: Results of BERT-base across two different loss functions, that is, cosine, InfoNCE, and their combination cosine + InfoNCE.

	MRR	Hits@1	Hits@3	Hits@10
Raw ($\hat{h}_o = h_r + h_s$)	0.0474	0.0245	0.0486	0.0881
cos	0.2417	0.1730	0.2600	0.3780
InfoNCE	0.3174	0.2286	0.3538	**0.5232**
cos +InfoNCE	**0.3314**	**0.2402**	**0.3554**	**0.5232**

B Layer Selection Details

Table 5 compares the chosen layer with the last layer under the same linear map with a contrastive training objective. This measures the benefit of selecting a layer based on the validation set on this task as opposed to alternative more complex layer-selection methods. Table 6 presents per-model MRR uncertainty for the linear map and MoT with contrastive objective initialized using three different seeds 0, 1, and 2. Mean and standard deviation are reported over different metrics.

Table 5: Comparative results of MRR on selected layers based on validation results as opposed to the last layer. Only for Granite model, there is no difference between the last and the selected layer.

Model	Selected / total layers	Selected	Last Layer	Difference
Gemma-4-E2B	27/35	0.410	0.375	+0.035
Gemma-4-E4B	33/42	0.447	0.434	+0.013
Gemma-4-E4B-it	5/42	0.273	0.256	+0.017
Gemma-4-12B	39/48	0.463	0.423	+0.040
Granite-4.2-8B	40/40	0.295	0.295	+0.000
Llama-3.1-8B	25/32	0.497	0.464	+0.034
Llama-3.1-8B-Instruct	23/32	0.493	0.464	+0.029
Llama-3.2-1B	11/16	0.427	0.426	+0.001
Llama-3.2-3B	19/28	0.457	0.453	+0.004
Ministral-3-8B-Instruct	27/34	0.439	0.424	+0.015
ModernBERT-base	3/22	0.269	0.257	+0.012
OLMo-3-7B	27/32	0.423	0.410	+0.014
OLMo-3-7B-Instruct	23/32	0.371	0.363	+0.007
Qwen3.5-0.8B	21/24	0.358	0.346	+0.013
Qwen3.5-2B	21/24	0.387	0.375	+0.012
Qwen3.5-4B	26/32	0.428	0.415	+0.012
Qwen3.5-4B-Base	28/32	0.436	0.421	+0.015
Qwen3.5-9B	28/32	0.470	0.442	+0.028
SmolLM3-3B	35/36	0.345	0.333	+0.012
Mean MRR		0.405	0.388	0.016

The selected layers for all 19 models are used for the linear map and MoT. To test whether this layer selection explains the smaller MRR gain with MoT, we let all four configurations select layers independently on a bounded set of models and graphs. BERT-base is evaluated on Wikidata and ConceptNet; Qwen3.5-4B and Gemma-4-E4B are evaluated on Wikidata. The comparison includes BERT with ConceptNet because its common-layer objective pattern differs from Wikidata's.

Each configuration evaluates at the same five layers within a model: 2, 4, 6, 10, and 12 for BERT; 6, 12, 20, 26, and 32 for Qwen; and 9, 18, 24, 33, and 42 for Gemma. BERT uses mean pooling

Table 6: Per-model MRR uncertainty for the linear map and MoT with contrastive objective initialized using three different seeds 0, 1, and 2. Mean and standard deviation of MRR are reported.

Model	Layer	Linear Map (mean MRR ± SD)	MoT (mean MRR ± SD)
Gemma-4-E2B	27/35	0.410 ± 0.000	0.417 ± 0.002
Gemma-4-E4B	33/42	0.447 ± 0.001	0.453 ± 0.004
Gemma-4-E4B-it	5/42	0.273 ± 0.001	0.281 ± 0.001
Gemma-4-12B	39/48	0.463 ± 0.000	0.473 ± 0.002
Granite-4.2-8B	40/40	0.295 ± 0.001	0.319 ± 0.002
Llama-3.1-8B	25/32	0.497 ± 0.001	0.503 ± 0.001
Llama-3.1-8B-Instruct	23/32	0.493 ± 0.001	0.499 ± 0.002
Llama-3.2-1B	11/16	0.427 ± 0.001	0.435 ± 0.002
Llama-3.2-3B	19/28	0.457 ± 0.000	0.469 ± 0.002
Ministral-3-8B-Instruct	27/34	0.439 ± 0.002	0.449 ± 0.001
ModernBERT-base	3/22	0.269 ± 0.001	0.300 ± 0.003
OLMo-3-7B	27/32	0.423 ± 0.000	0.435 ± 0.001
OLMo-3-7B-Instruct	23/32	0.371 ± 0.000	0.389 ± 0.003
Qwen3.5-0.8B	21/24	0.358 ± 0.001	0.366 ± 0.004
Qwen3.5-2B	21/24	0.387 ± 0.001	0.400 ± 0.003
Qwen3.5-4B	26/32	0.428 ± 0.002	0.434 ± 0.003
Qwen3.5-4B-Base	28/32	0.436 ± 0.002	0.440 ± 0.002
Qwen3.5-9B	28/32	0.470 ± 0.001	0.476 ± 0.002
SmolLM3-3B	35/36	0.345 ± 0.000	0.376 ± 0.004

Table 7: Independent validation-layer selection on a preselected five-depth grid (on five pre-selected layers). ℓ_0 is the fixed layer and $\hat{\ell}$ is the validation-selected layer with maximum MRR. MRR entries are mean ± SD for full-training of maps initialized three times with different seeds (=0, 1, and 2). Thus, mean and SD are calculated over three MRR values. Δ denotes difference in MRR between the selected layer and the fixed layer, chosen as reference. Intervals are paired bootstrap 95% intervals over test triples after averaging mean reciprocal rank across seeds. Selected layer finally used only one map initialized using seed 1729.

Setting	Variant	ℓ_0	$\hat{\ell}$	MRR (fixed layer)	MRR (selected)	Δ [95% CI]
BERT / Wikidata	single / cos.	10	10	0.273 ± 0.001	0.273 ± 0.001	+0.000 [+0.000, +0.000]
	mix. / cos.	10	10	0.323 ± 0.006	0.323 ± 0.006	+0.000 [+0.000, +0.000]
	single / comp.	10	6	0.345 ± 0.001	0.338 ± 0.003	-0.008 [-0.014, -0.001]
	mix. / comp.	10	10	0.360 ± 0.002	0.360 ± 0.002	+0.000 [+0.000, +0.000]
BERT / ConceptNet	single / cos.	10	4	0.060 ± 0.001	0.089 ± 0.002	+0.029 [+0.022, +0.035]
	mix. / cos.	10	4	0.086 ± 0.005	0.119 ± 0.002	+0.033 [+0.026, +0.040]
	single / comp.	10	4	0.194 ± 0.001	0.209 ± 0.004	+0.015 [+0.008, +0.022]
	mix. / comp.	10	4	0.232 ± 0.002	0.244 ± 0.001	+0.012 [+0.005, +0.019]
Qwen3.5-4B / Wikidata	single / cos.	26	26	0.324 ± 0.000	0.324 ± 0.000	+0.000 [+0.000, +0.000]
	mix. / cos.	26	26	0.383 ± 0.003	0.383 ± 0.003	+0.000 [+0.000, +0.000]
	single / comp.	26	26	0.429 ± 0.001	0.429 ± 0.001	+0.000 [+0.000, +0.000]
	mix. / comp.	26	26	0.431 ± 0.001	0.431 ± 0.001	+0.000 [+0.000, +0.000]
Gemma-4-E4B / Wikidata	single / cos.	33	33	0.287 ± 0.001	0.287 ± 0.001	+0.000 [+0.000, +0.000]
	mix. / cos.	33	33	0.377 ± 0.004	0.377 ± 0.004	+0.000 [+0.000, +0.000]
	single / comp.	33	33	0.446 ± 0.000	0.446 ± 0.000	+0.000 [+0.000, +0.000]
	mix. / comp.	33	33	0.453 ± 0.001	0.453 ± 0.001	+0.000 [+0.000, +0.000]

and the decoders use the last token. The experiments are depicted in Fig. 5 and Table 7. Selection trains on the same 8,000-row calibration sample with seed 1729 and maximizes MRR over the full validation split. Layer selection uses validation ranking only; filtering uses known validation objects from all three splits, as in the main protocol.

After selection, each configuration is trained on the full training split with seeds 0, 1, and 2. Its fixed-layer reference is trained with the same implementation and seeds, that is, layer 10 for BERT, 26 for Qwen, and 33 for Gemma. Both selected and fixed layers reuse the same trained linear map rather than being considered independent repetitions. Training uses the main cosine and composite

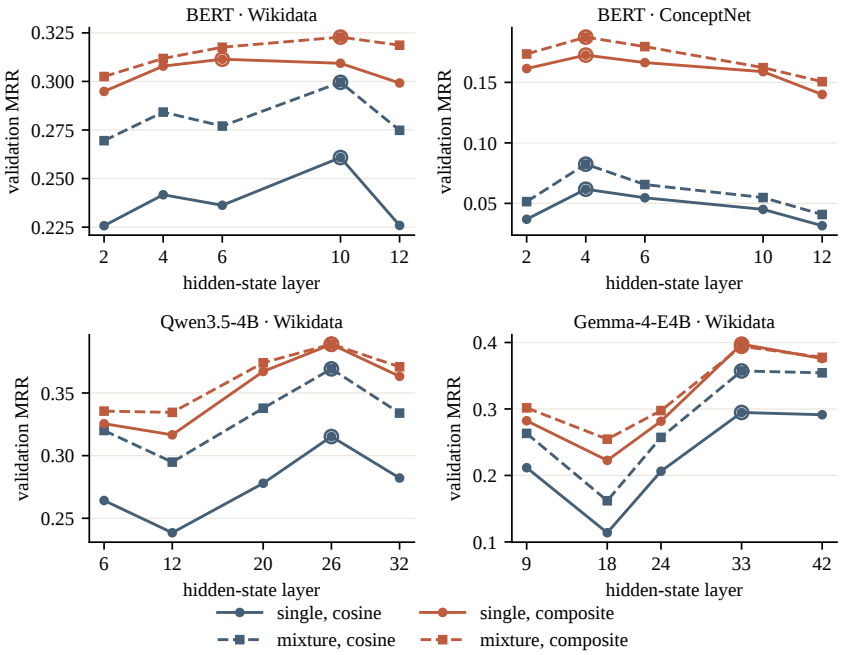


Figure 5: Equal validation searches for four configurations: single linear map trained with cosine loss, single linear map trained with composite loss, MoT or mixture of maps trained with cosine loss, and MoT trained with the composite loss. Each of the subgraphs show validation MRR at five pre-selected layers. Each configuration selects its own maximum before final training and test evaluation. For this experiment, linear maps were initialized only once with a seed and used a subset of 8,000 training examples, while test results presented in Table 7 used the full training split and mean MRR using maps initialized three times with different seeds. This is to note that these coarse grids can miss better intervening layers.

contrastive losses, temperature 0.05, AdamW with learning rate 0.001 and weight decay 0.0001, batch size 512, at most 60 epochs, and early stopping after 12 unimproved validation loss checks. MoT keeps its 0.01 balancing penalty.

On Qwen and Gemma, all four configurations independently select the common reference layer. On BERT with Wikidata, the linear map with InfoNCE selects layer 6, while the other configurations select layer 10, however, the difference is -0.0075 in MRR. Selecting by validation therefore does not guarantee an improvement on test. On BERT with ConceptNet, all configurations select layer 4 and improve by 0.012–0.033 MRR over layer 10. As the experiments with Wikidata are more robust in the sense of higher MRRs, we continue the layer selection for the other test and all other models with Wikidata triples.

C Ablation Study on Linear Map

Table 8 presents detailed ablation on linear maps wherein each row removes one possible explanation for the linear map's score and reports the MRR that remains.

D Alignment capacity

We test the importance of the triple size for the linear alignment in the process of transferring the linear map from one model to another. The maximum number of composed subject-relation representations corresponds to the training data size of 33,937. The importance of data size and flexibility is tested across six directed pairs of models as shown in Fig. 6. Qwen3.5-4B is the reference for queries from Qwen3.5-0.8B, Qwen3.5-9B, SmolLM3-3B, and Llama-3.1-8B. Llama-3.1-8B is the reference for ModernBERT and Gemma-4-12B. These pairs cover within- and cross-family transfer, several model sizes, and one encoder; they are a selected diagnostic set, not a random sample of the

Table 8: Each row removes one possible explanation for the linear map's score and reports the MRR that remains. The upper block uses BERT-base on the standard Wikidata split, where the trained map scores 0.355. The lower block uses a split in which no test subject or object string appears in training, and compares two training losses for the MoT.

Setting	What changes	MRR
BERT-base, standard Wikidata split		
Trained map	Frozen pretrained BERT-base, map trained on the true triples. The controls below are compared with this score.	0.355
Random targets	Same map and training, but the object labels are randomly reassigned, so there is no real fact to learn.	0.163
Random encoder	BERT-base with random weights instead of pretrained ones. The shaping map is trained exactly as above.	0.226
Majority object	No encoder or map. For each composed vector, objects are ranked by how often they represent the gold-truth in training.	0.217
Character trigrams	No encoder or map. Objects are ranked by how many three-character sequences they share with the subject and relation text.	0.072
Fine-tuned BERT	BERT-base itself is fine-tuned with a SimKGC-style objective on the same split and candidate pool. This is a reference for what updating the model gains, not a control.	0.381
Dataset with unique triples		*contrastive / cosine*
Qwen3.5-4B	Eight-map MoT trained with the contrastive loss, and the same MoT trained with cosine loss only.	0.329 / 0.305
SmolLM3-3B	As above.	0.252 / 0.228
Gemma-4-E4B	As above.	0.313 / 0.260

Table 9: Alignment capacity and sample efficiency. Entries at 512, 2,048, 8,192, and 33,937 are MRR divided by the same pair's full-data full-ridge MRR. The first three columns report mean $\pm$ standard deviation over three nested-subset permutations; the full-data column is one deterministic fit. Truncated ridge retains leading source singular directions and is not response-aware reduced-rank regression. The final column gives absolute full-data MRR.

(a) Qwen3.5-4B $\leftarrow$ Qwen3.5-0.8B
(b) Qwen3.5-4B $\leftarrow$ Qwen3.5-9B
(c) Qwen3.5-4B $\leftarrow$ SmolLM3-3B
(d) Qwen3.5-4B $\leftarrow$ Llama-3.1-8B
(e) Llama-3.1-8B $\leftarrow$ ModernBERT-base
(f) Llama-3.1-8B $\leftarrow$ Gemma-4-12B

Target $\leftarrow$ source	Map	512	2,048	8,192	33,937	Full MRR
(a)	Full ridge	0.705 ± 0.014	0.936 ± 0.005	0.993 ± 0.006	1.000	0.364
	Trunc. 256	0.837 ± 0.005	0.922 ± 0.005	0.944 ± 0.006	0.942	0.343
	Trunc. 64	0.686 ± 0.007	0.734 ± 0.007	0.748 ± 0.002	0.751	0.273
	Orthogonal	0.815 ± 0.011	0.925 ± 0.005	0.954 ± 0.004	0.965	0.352
(b)	Full ridge	0.791 ± 0.003	0.970 ± 0.002	0.995 ± 0.002	1.000	0.436
	Trunc. 256	0.852 ± 0.006	0.917 ± 0.003	0.929 ± 0.004	0.938	0.409
	Trunc. 64	0.637 ± 0.006	0.671 ± 0.014	0.681 ± 0.004	0.680	0.297
	Orthogonal	0.863 ± 0.009	0.981 ± 0.000	0.993 ± 0.001	0.996	0.434
(c)	Full ridge	0.752 ± 0.017	0.941 ± 0.009	0.994 ± 0.002	1.000	0.379
	Trunc. 256	0.802 ± 0.016	0.897 ± 0.007	0.925 ± 0.003	0.940	0.356
	Trunc. 64	0.634 ± 0.016	0.691 ± 0.010	0.705 ± 0.005	0.712	0.270
	Orthogonal	0.735 ± 0.006	0.865 ± 0.003	0.909 ± 0.004	0.926	0.351
(d)	Full ridge	0.745 ± 0.011	0.950 ± 0.007	0.991 ± 0.003	1.000	0.442
	Trunc. 256	0.832 ± 0.009	0.911 ± 0.009	0.928 ± 0.001	0.937	0.414
	Trunc. 64	0.612 ± 0.003	0.657 ± 0.012	0.660 ± 0.003	0.668	0.295
	Orthogonal	0.822 ± 0.008	0.950 ± 0.004	0.978 ± 0.001	0.987	0.436
(e)	Full ridge	0.748 ± 0.042	0.933 ± 0.008	0.979 ± 0.007	1.000	0.271
	Trunc. 256	0.734 ± 0.036	0.876 ± 0.006	0.911 ± 0.014	0.929	0.252
	Trunc. 64	0.601 ± 0.052	0.666 ± 0.015	0.683 ± 0.002	0.692	0.187
	Orthogonal	0.011 ± 0.000	0.011 ± 0.000	0.011 ± 0.000	0.012	0.003
(f)	Full ridge	0.779 ± 0.009	0.959 ± 0.004	0.994 ± 0.003	1.000	0.471
	Trunc. 256	0.815 ± 0.007	0.902 ± 0.008	0.927 ± 0.003	0.932	0.439
	Trunc. 64	0.578 ± 0.002	0.627 ± 0.009	0.644 ± 0.003	0.650	0.306
	Orthogonal	0.746 ± 0.005	0.889 ± 0.003	0.918 ± 0.004	0.936	0.441

model set. ModernBERT uses the historical last-token representations from the transfer study, not the mean-pooled representations in the main ranking table.

For each model pair, the reference linear map, candidate vectors, and 4,243 test triples stay fixed. Only the alignment changes. We compare full ridge, ridge restricted to the leading 64 or 256 principal directions of the input composed vectors, and an orthogonal alignment. The restricted maps keep directions with the greatest input variance; they do not choose directions by how well they predict the reference composed vector. The orthogonal map preserves lengths and angles, so it tests whether a rotation or reflection suffices. All maps omit an intercept and additional centering or rescaling. Ridge minimizes mean squared reconstruction error with squared-weight penalty 0.01. The input representations retain their original training-fitted PCA transformation where applicable.

The experiments presented in Fig. 6 compare training size subsets of 512, 2,048, and 8,192 with the full training set size to measure the approach's sensitivity to alignment examples. Before the comparison, full-ridge fits reproduce the original transfer scores within 0.00002 MRR on every pair.

The difference of the data size is negligible for the last subset and the full dataset size, however, the MRR drops considerably for all transfers with only 512 or 2,048 samples. With a very small data size of 512, the truncated ridge or ridge restricted to 256 principal directions of the input composed vector outperforms the full ridge. At 2,048 examples this ratio drops to 0.954, and with all examples to 0.936 (Table 9). Restricting the map can therefore help with little alignment data. In terms of model pairing, all of the experimented pairings show that the knowledge transfer leads to a successful performance with a sufficient number examples, without major deviations or outliers.

When we transfer representations into a reference model's space, they can work quite well. However, bringing the resulting prediction back to the source model's object space requires another learned alignment. Here, the return transfer can be viewed as $x_B \xrightarrow{S_{B\to A}} \tilde{x}_A \xrightarrow{W_A} \hat{h}_o^A \xrightarrow{S_{A\to B}} \tilde{h}_o^B$, where the additional alignment $S_{A\to B}$ maps the object prediction back to the LM B's space. A reverse map fitted on composed vectors retains a median 64% of the input model's (LM B) native-linear map MRR across the 18 non-Llama-3.1-8B models, considering LM A to be Llama-3.1-8B. Fitting the reverse map on the corresponding object vectors raises this to 80%. The improvement suggests that the composed vector and object vector representations are geometrically different. Since even the object-based mapping does not fully recover native performance, the two representation spaces cannot simply be treated as interchangeable. Thus, successful transfer into a reference space does not imply that the transformation can be perfectly reversed.

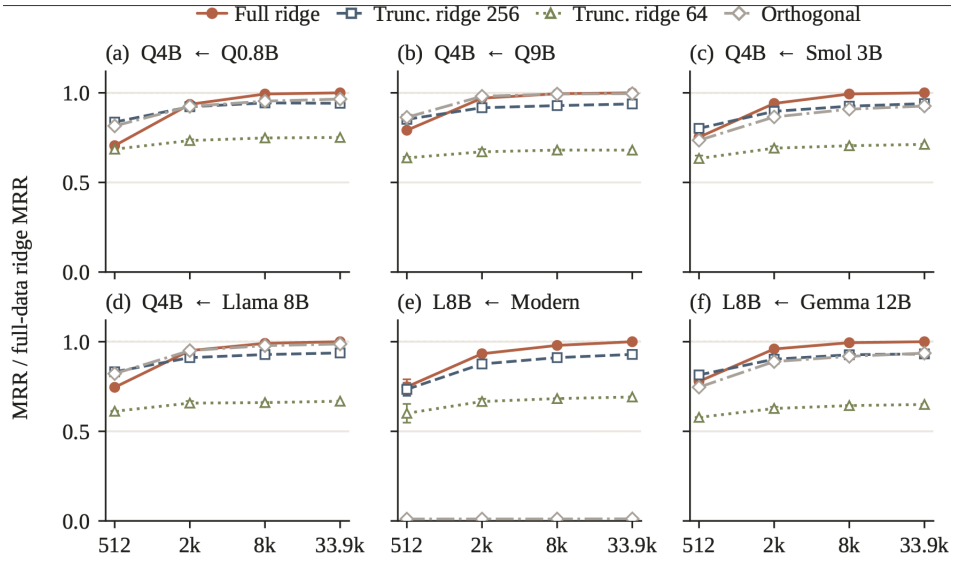


Figure 6: Impact on alignment data size on knowledge transfer performance for paired models based on MRR. x-axis displays the number of training pairs used for training the maps.

Table 10: Model indices shared by the transfer and fact-agreement figures. The transfer matrix uses models 1–19; fact agreement additionally includes Falcon-Mamba.

Index	Model	Index	Model
1	Qwen3.5-0.8B	11	Ministral-3-8B
2	Qwen3.5-2B	12	Llama-3.1-8B
3	Qwen3.5-4B	13	Llama-3.1-8B-Instruct
4	Qwen3.5-4B-Base	14	Granite-4.2-8B
5	Qwen3.5-9B	15	Llama-3.2-1B
6	SmolLM3-3B	16	Llama-3.2-3B
7	Gemma-4-12B	17	ModernBERT-base
8	Gemma-4-E2B	18	OLMo-3-7B
9	Gemma-4-E4B	19	OLMo-3-7B-Instruct
10	Gemma-4-E4B-it	20	Falcon-Mamba-7B

Table 11: Architecture, published size, and measured linear map-MRR change for the seven base/post-training pairs. The drop groups are descriptive bins, not learned clusters. The parameter conventions follow the model cards and therefore differ for Gemma E2B/E4B, whose cards distinguish effective parameters from parameters including embeddings.

Drop group	Pair	Architecture	Published size	ΔMRR
Small	Llama-3.1-8B	Autoregressive transformer with grouped-query attention	8B	-0.0036
Small	Qwen3.5-4B	Causal LM with vision encoder; hybrid Gated DeltaNet/attention layout	4B	-0.0053
Moderate	OLMo-3-7B	Decoder-only transformer (32 layers)	7B	-0.0462
Moderate	OLMo-Hybrid-7B	Hybrid causal LM: three Gated DeltaNet layers per attention layer	7B	-0.0437
Large	Gemma-4-E2B	Dense multimodal decoder; hybrid local/global attention with per-layer embeddings	2.3B effective; 5.1B with embeddings	-0.1496
Large	Gemma-4-E4B	Dense multimodal decoder; hybrid local/global attention with per-layer embeddings	4.5B effective; 8B with embeddings	-0.1722
Large	Gemma-4-12B	Dense, encoder-free unified multimodal decoder; hybrid local/global attention	11.95B	-0.2577

E Accessibility and Fact-Level Agreement Details

From Table 11, we observe that the mean-MRR drop is small for the matched Llama and Qwen pairs (about 0.004 and 0.005), intermediate for OLMo-3 and OLMo-Hybrid (about 0.045 each), and larger for the three Gemma pairs (0.150–0.258). For the two OLMo models with intermediate checkpoints, most of the drop occurs during supervised fine-tuning; later stages change the score much less. These comparisons describe the measured model pairs. Because architecture, training data, and post-training procedure vary together, they do not identify which factor causes the differences.

Figure 7 compares the paired endpoints and intermediate training stages. The seven matched base/post-trained pairs allow a comparison within model families. Table 12 reports three different measurements. Linear map MRR measures supervised object prediction with the eight-map MoT at each model's selected layer. The cost column measures perplexity change under the separate intervention-cost protocol. They should not be treated as three independent estimates of a single quantity.

The linear map MRR is lower after post-training in all seven point estimates, but the differences are highly uneven. The Llama and Qwen changes are small, whereas the three Gemma pairs lose 0.150 to 0.258 MRR. Intervention cost rises on only four pairs. Thus the data support a family-dependent change in accessibility, not a general claim that post-training makes every intervention

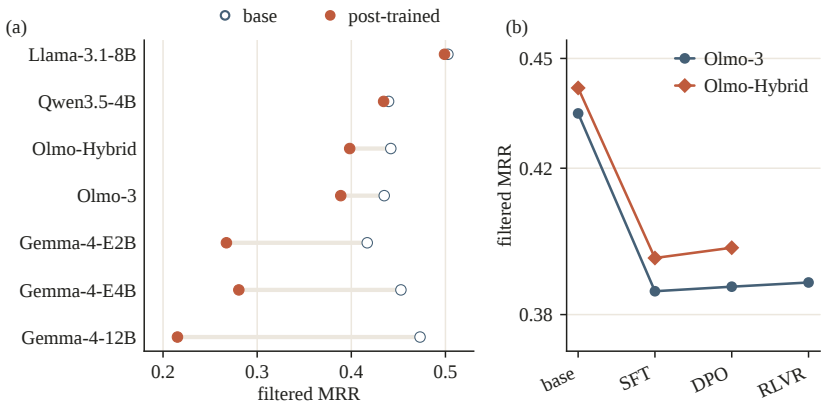


Figure 7: Post-training changes the linear map's access to relational information unevenly. Left: matched base and post-trained linear map MRR for seven pairs. Right: intermediate stages for OLMo-3 and OLMo-Hybrid. SFT denotes supervised fine-tuning, DPO direct preference optimization, and RLVR reinforcement learning with verifiable rewards. Each model uses its recorded linear map protocol and selected layer. These measurements concern the trained linear map, not the model's general factual capability.

Table 12: Matched base and post-trained models. The MRR columns show linear map prediction; cost is the percentage change in perplexity under the separate intervention protocol, not the change caused by post-training alone.

Model pair	Base MRR	Post MRR	Base cost	Post cost
Llama-3.1-8B	0.503	0.499	+284.7%	+551.4%
Olmo-3-7B	0.435	0.389	+29.0%	+45.3%
Gemma-4-E4B	0.453	0.281	+5.0%	+26.4%
Qwen3.5-4B	0.440	0.434	+31.3%	+33.7%
Gemma-4-E2B	0.417	0.267	+5.6%	−16.0%
Gemma-4-12B	0.473	0.215	+197.5%	+10.4%
Olmo-Hybrid-7B	0.442	0.398	+15.0%	+2.4%

more costly. Shared families and small seed budgets also limit an inference from the seven signs. Large random-relative effects need caution as the Gemma-4-12B base gain of 11.134 nats does not establish improvement over the untouched model, because the random intervention itself can severely damage the prediction.

The intermediate checkpoints provide more detail for OLMo-3 and OLMo-Hybrid, two 7B architectures trained by the same organization. In both, most of the base-to-final change in linear map MRR occurs at supervised fine-tuning; subsequent preference or reinforcement stages change it much less. For OLMo-3, MRR goes from 0.435 at base to 0.386 after supervised fine-tuning, then 0.388 after preference optimization and 0.389 in the final model. These observations localize the change within these training procedures. They do not show that supervised fine-tuning generally erases relational knowledge.

Figure 8 separates overall model quality, relation difficulty, and their remaining interaction. The fact-agreement analysis in Section 4.3 compares the ordering of individual test examples. A complementary analysis asks whether different models specialize in different relation types. Let E_{mr} be model m's mean reciprocal rank on relation r. We fit a model effect and a relation effect, leaving an interaction residual:

$$E_{mr} = a_m + b_r + I_{mr}. \tag{8}$$

The fit weights each relation by its share of the evaluated triples. Small residual variance means that knowing a model's average performance and a relation's average difficulty explains much of that table. It does not mean that all models answer the same individual facts.

On the 62 Wikidata relations with at least ten test examples, interactions account for 4.1% of the weighted variance. Even an oracle selecting the best of the 19 models separately for each relation

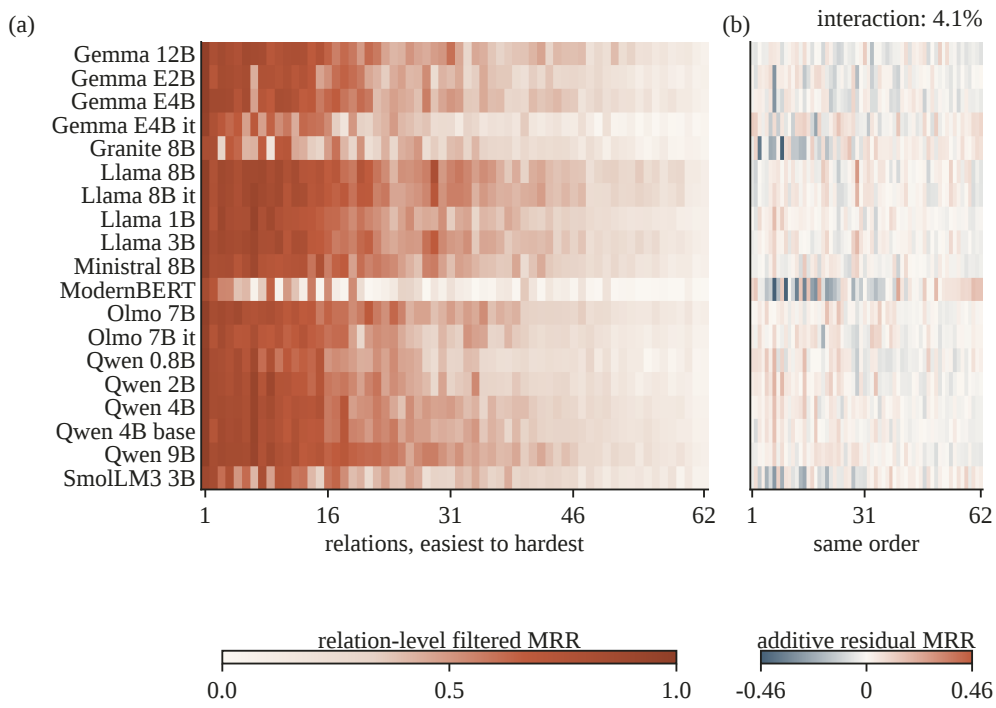


Figure 8: Model quality and relation difficulty explain much of the relation-level performance matrix. Left: MRR for 19 models on 62 Wikidata relations. Right: the residual after removing weighted model and relation effects. Both panels use the same row and column order; the residual uses a zero-centered scale. Interactions account for 4.1% of weighted variance, although individual model–relation exceptions remain.

improves MRR only from 0.493 to 0.514. This is an oracle over whole relations, not over individual queries; it does not bound a query-dependent ensemble. On nine ConceptNet relations, the analogous gain is 0.011. The learned relation router is evaluated separately on 14 held-out Wikidata relations: it reaches 0.387, compared with 0.391 for the best single model and 0.418 for the relation oracle. A richer routing method might improve the learned router, but the measured relation-wise oracle limits the gain available in this setting.

Tables 11 place the measured post-training changes beside the public descriptions of those models. The metadata provides context for the observed differences, not a causal explanation.

Table 13 asks whether the linear map finds facts that prompting the same frozen model cannot. We use the same 1,000 Wikidata test triples. The result of a prompt is scored by its mean token log probability not penalizing long names. The utilized prompting strategies are zero-shot with a fixed template per relation, five-shot, paraphrase, and soft prompt. An example of a zero-shot prompt would be for *country*, *place of birth*, and *occupation* the templates are "s is located in the country of", "s was born in", and "The occupation of s is". Thirty relations have hand-written templates, and the rest use "s: r is". The *paraphrase* strategy picks one of five templates per relation using at most 20 validation examples, shuffled with seed zero. Relations without validation examples keep the default template. The *five-shot* strategy places five training examples of the same relation before the question, each written as a completed prompt followed by a period and shuffled with seed one. The *soft-prompt* strategy trains five continuous prompt vectors per relation on the shaping map's labeled training triples while the model stays frozen. The most successful strategy is shown in Table 13 for each mode. The best prompting strategy comes close in performance to the linear map and only the soft prompt for Llama outperforms the linear map. The right half of the table tests how many facts are missed in the sense of normalized output of a zero-shot prompt. Most facts are missed: 900 of 1,000 for Qwen3.5-4B, 874 for Gemma-4-E4B, and 862 for Llama-3.1-8B. These columns rescore the linear map, the best prompt, and the majority-object baseline on the missed facts only.

Table 13: Linear map MRR performance compared to object strings obtained from prompting the same frozen models. The most successful prompting strategy out of paraphrase, five-shot and soft prompt is listed here. The right side shows objects each model fails to predict with the zero-shot prompt.

	All 1,000 facts				Facts missed by the model			
Model	Linear Map W	Zero-shot	Best prompt	Best Strategy	Missed	Linear Map W	Best prompt	Majority
Qwen3.5-4B	0.449	0.151	0.412	soft prompt	900	0.409	0.357	0.215
Gemma-4-E4B	0.462	0.210	0.438	five-shot	874	0.426	0.376	0.212
Llama-3.1-8B	0.510	0.231	0.565	soft prompt	862	0.479	0.509	0.221

The zero-shot column on the left is a ranking score rather than a generation test, so a fact can be missed in generation even when its correct object ranks near the top.