AlignSAE: Concept-Aligned Sparse Autoencoders

Anonymous authors
Paper under double-blind review

Abstract

Large Language Models (LLMs) encode factual knowledge within hidden parametric spaces that are difficult to inspect or control. While Sparse Autoencoders (SAEs) can decompose hidden activations into more fine-grained, interpretable features, they often struggle to reliably align these features with human-defined concepts, resulting in entangled and distributed feature representations. To address this, we introduced AlignSAE (Anonymous Repo) , a method that aligns SAE features with a predefined ontology through a "pre-train, then post-train" curriculum. After an initial unsupervised training phase, we apply supervised post-training to bind specific concepts to dedicated latent slots while preserving the remaining capacity for general reconstruction. This separation creates an interpretable interface where specific concepts can be inspected and controlled without interference from unrelated features. Empirical results demonstrate that AlignSAE enables precise causal interventions, such as reliable "concept swaps", by targeting single, semantically aligned slots, and further supports multi-hop reasoning and a mechanistic probe of grokking-like generalization dynamics.

1 Introduction

While Large Language Models (LLMs) have advanced rapidly, their internal mechanisms remain opaque. Mechanistic interpretability (Bereska & Gavves, 2024; Huben et al., 2024) seeks to bridge this gap by reverse-engineering these models and decomposing their internal computations into interpretable components. Early efforts focused on inspecting individual neurons (Nguyen et al., 2019; Elhage et al., 2022; Bills et al., 2023), operating under the assumption that specific neurons would map one-to-one onto human concepts. However, this approach faced a fundamental barrier of superposition, *i.e.*, neural networks represent more independent features than available neurons by encoding each feature as a linear combination of neurons (Ferrando et al., 2024). Consequently, single neurons become difficult to interpret, as their activations represent entangled mixtures of concepts.

This limitation of neuron-level analysis directly motivated the development of *Sparse Autoencoders (SAEs)*. The idea is to disentangle these superimposed neurons into more interpretable *features*, by learning an overcomplete, sparse representation of neural activations (Shu et al., 2025). By mapping LLM's hidden states into higher-dimensional space, SAEs often learned features that are cleaner and more interpretable than individual neurons (Leask et al., 2025; Chanin et al., 2025; Yan et al., 2025).

Ideally, features learned by an SAE would correspond to atomic, independent, human-interpretable concepts, so that a human can easily inspect, interpret, and steer model behavior. For example, in the context of relation extraction, one would expect a single SAE feature to exclusively represent the relation Birth_City, such that manipulating this feature alone would precisely control the model's output regarding cities of birth. However, because standard SAEs are trained in an unsupervised fashion, they *have no explicit incentive to align their latent features with human-defined concepts*. In practice, this leads to two major challenges in interpreting the SAE feature space: 1) *Feature interpretation is non-trivial*: determining which feature corresponds to a target concept is difficult. To infer a feature's semantics, practitioners often rely on constructing minimal contrast pairs (Jing et al., 2025; Li et al., 2025) or inspecting top-activating examples (Cunningham et al., 2023; Bereska & Gavves, 2024; Shu et al., 2025). 2) *Features remain entangled*: concepts are often fragmented

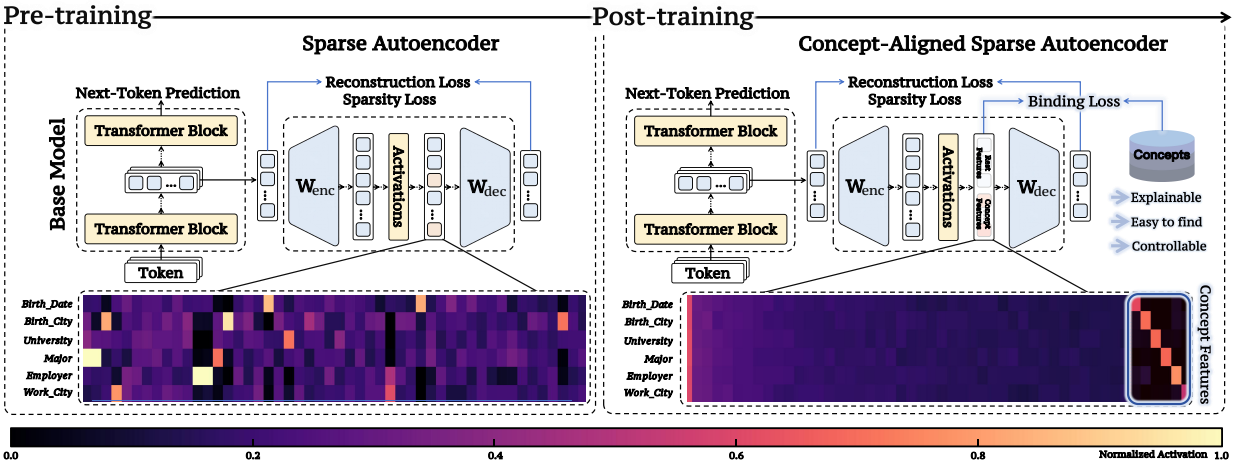


Figure 1: Overview of AlignSAE. Left: An unsupervised SAE trained post hoc on frozen LLM activations optimizes only reconstruction and sparsity, so each concept tends to be spread across multiple features, making interventions unreliable. Right: Our AlignSAE adds a supervised binding loss that maps each concept in a pre-defined ontology to a dedicated feature, yielding clean, isolated activations that are easy to find, interpret, and control.

across multiple features, and conversely, a single feature may correspond to multiple unrelated concepts, as shown in Fig. 1 (left). These limitations undermine downstream applications that require reliable feature-level control, such as safety steering (Bereska & Gavves, 2024; Bhattacharjee et al., 2024; Ghosh et al., 2025; O'Brien et al., 2025), knowledge editing (Makelov et al., 2024; Guo et al., 2024; Farrell et al., 2024; Zhao et al., 2025; Karvonen et al., 2025), and data attribution (He et al., 2024; Muhamed et al., 2025).

To mitigate these issues, we take inspiration from the training pipeline of LLMs. As illustrated in Fig. 1, we view conventional SAE training as analogous to LLM pre-training: an unsupervised phase that discovers a broad latent feature space but does not guarantee alignment with human concepts. In LLMs, this misalignment is addressed by post-training steps such as instruction tuning (Wei et al., 2022; Zhang et al., 2025) or RLHF (Ouyang et al., 2022). By analogy, we propose an **SAE post-training** stage that guides the unsupervised SAE to align the SAE's feature space with a set of chosen concepts, turning it from a reconstructive tool into a reliable concept-level interface.

Concretely, we attach an SAE to one layer of a frozen base LM and train it in two phases: first unsupervised, then supervised. After the SAE has learned to reconstruct the activations, we "fine-tune" the SAE with concept supervision. We reserve $|\mathcal{R}|$ dedicated concept slots (one per ontology concept), while the remaining K dimensions form a free feature bank to preserve reconstruction fidelity. We then augment the training objective with additional losses to bind and isolate each concept in its corresponding feature slot. In particular, we impose: (i) a *concept binding loss* that forces a one-to-one mapping between each labeled concept and a dedicated feature, (ii) a *concept invariance loss* that makes each concept feature invariant to irrelevant variations and decorrelates it from the free features, and (iii) a *sufficiency loss* that trains an auxiliary answer head to rely only on the concept slots for predicting concept-related information. Together, these objectives encourage the encoder to route concept-specific evidence into the appropriate slot rather than dispersing it across the latent space. The result is an SAE feature space that directly corresponds to human-interpretable concepts, as shown in Fig. 1 (right). In this work, we focus on the important NLP task of relation completion (RC). Accordingly, we use an ontology where concepts correspond to relation types, *e.g.*, `BIRTH_CITY`.[1]

Empirically, AlignSAE is more interpretable, disentangled, and controllable than a standard SAE under three RC evaluations. **(i) Binding/generalization:** concept–slot binding becomes clean and diagonal at mid layers (Fig. 3) and transfers to unseen templates (Fig. 4). **(ii) Disentanglement:** concepts concentrate onto single slots with low fragmentation and high Top-1 mass (Fig. 5). **(iii) Causal control:** swap interventions

[1]In principle, AlignSAE should be applicable to any type of ontology. We leave this additional study as future work.

exhibit a clear fidelity–strength tradeoff (Table 1), strong mid-layer controllability (Fig. 6), and predictable failure modes (Table 2); in 2-hop RC, post-trained AlignSAE attains 4× higher swap success than the traditional SAE (Fig. 8).

2 Related Work

We review two main directions that influenced this work: *Sparse Autoencoder Steering* and *Concept Binding*, which address interpretable control and concept alignment, respectively. Inspired by both, we introduce AlignSAE, a lightweight, concept-aligned, and interpretable SAE framework.

Sparse Autoencoder Steering. Sparse Autoencoders (SAEs) provide an interpretable interface to LLM activations by decomposing superposed, polysemantic neuron activity into sparse, overcomplete features (Bricken et al., 2023; Cunningham et al., 2023). This representation enables *SAE steering*, where intervening on specific features can causally influence model outputs toward desired behaviors or concepts (O'Brien et al., 2025; Marks et al., 2025; Arad et al., 2025). Recent work improves feature quality through training objectives and architectural choices (Rajamanoharan et al., 2024; Bricken et al., 2023; Shu et al., 2025; Sharkey et al., 2025). However, since SAE features are learned in a purely unsupervised manner, they are not guaranteed to align with a user-specified concept set: a target concept may be fragmented across multiple features, and individual features may mix unrelated signals. As a result, practical steering often still relies on manual feature identification, including contrast pairs and feature search heuristics (O'Brien et al., 2025; Jing et al., 2025; Bayat et al., 2025; Chalnev et al., 2024; Yang et al., 2025), which are limited due to the remaining poly-semanticity of the features (Chanin et al., 2025; Cui et al., 2025; Minegishi et al., 2025). To address this, we post-train an SAE with concept supervision to learn a stable concept–feature mapping: we allocate dedicated concept slots, bind each ontology relation to a fixed slot, and make steering interventions directly targetable without post-hoc feature search. A concurrent work, G-SAE (Härle et al., 2025), trains label-supervised detectors with one sigmoid loss for monosemantic features, while AlignSAE post-trains with multiple objectives to yield ontology-aligned, step-wise slots, validated by causal swap interventions and 1-hop/2-hop relation completion.

Concept Binding. Posterior Regularization (PR) (Ganchev et al., 2010) and Logic Rule Encoding (LRE) (Hu et al., 2016) are traditional frameworks adopted to bind human-defined concepts to neural models by imposing soft constraints on posterior distribution, or integrating logic rules into the learning objectives. Prior work has applied PR to reading comprehension by enforcing linguistic concept-level constraints (Zhou et al., 2019), and to question answering by mapping event triggers to conceptual constraints (Lu et al., 2023). LRE works (Hu et al., 2016; Fischer et al., 2019; Yang et al., 2023) reconstruct the training objective by combining the task loss with a rule loss, thereby binding logical concepts to model predictions via parameter updates. Although LRE addresses the soft-constraint issue of PR, it remains a black-box mechanism that cannot be used to interpret or control specific internal representations of the model. Concept Bottleneck Models (Koh et al., 2020), originally proposed for image classification, map intermediate representations to human-readable concepts via a supervised concept-mapping loss. However, such heavy-weight intervention into the base model architecture is not easily scalable to large-scale models. To address these gaps, we propose a lightweight and interpretable framework, AlignSAE, which binds human-readable concepts to the intermediate representations of a frozen base model.

3 Method

We expose an LLM's implicit concept knowledge as an explicit, verifiable, and controllable interface via a *Concept-Aligned Sparse Autoencoder* (AlignSAE) trained on frozen-LLM activations.

Terminology. In this work, we focus on the task of relation completion (RC), where facts are represented as triples (e_1,r,e_2): the relation type r links entity mentions e_1 and e_2 (*e.g.*, (*"Marie Curie"*, BORN_IN, *"Warsaw"*)). In this paper, RC appears as *two tasks*. For 1-hop, given (e_1,r), the model predicts the missing object e_2; for instance, *"What is Alice's* `birth date`*?"* corresponds to a single relation query (§4.2.1). For 2-hop, given (e_1,r_1,r_2), the model predicts e_3 by composing two relations through an intermediate entity e_2, *i.e.*, $e_1 \xrightarrow{r_1} e_2 \xrightarrow{r_2} e_3$; *e.g.*, *"Who is the* `classmate` *of the* `mentor` *of*

Barbara?" (§4.2.2). Here we instantiate concepts as relation types from a domain ontology, a finite inventory of semantic links between entities (*e.g.*, BORN_IN, FRIEND_OF). We focus on relations because RC is a core NLP task with high-impact closed-domain applications (*e.g.*, biomedicine Demner-Fushman et al. (2025), finance (Grishman & Sundheim, 1996)), where the target relations are defined by an ontology; nevertheless, our framework is not tied to relations and could bind other ontology concepts (*e.g.*, entities or attributes) to supervised SAE slots.

Concept Binding in Activation Space. Given a frozen LLM M, we extract an activation h from layer ℓ and learn a sparse representation $z = E(h) \in \mathbb{R}^{|\mathcal{R}|+K}$ with an SAE $z = [z_{\text{concept}}; z_{\text{rest}}]$, where z_{concept} are $|\mathcal{R}|$ concept slots and z_{rest} are K free features ($|\mathcal{R}| \ll K$). We supervise z_{concept} so that each ontology concept occupies its own dedicated slot: for an input containing concept c, its slot should activate while other concept slots are suppressed. This makes concept–slot activations a direct, verifiable readout of the concept in use. We keep z_{rest} "free" to absorb residual linguistic variation and keep the base model's statistical structure.

A Verifiable Interface. Unlike a traditional SAE, which is primarily diagnostic, our approach constructs an operational interface that can be validated by intervention rather than relying on minimal comparison pairs. Specifically, (i) *verification*: we can check whether the model is using a particular relation by observing whether the corresponding concept slot activates; and (ii) *control*: we can causally steer the computation by manually activating or suppressing that slot, directly influencing the model's downstream prediction.

Model Training. Although AlignSAE is LLM-agnostic, we evaluate it on GPT-2[2]. We fine-tune the base model on our RC training data (§B), extract layer activations at the final question token across all layers, and train AlignSAE with a task-dependent free-feature bank of size K and $|\mathcal{R}|$ concept slots using a two-stage training (§C, 4.1).

SAE Pre-Training and Post-Training. Directly optimizing the full objective from scratch can produce unstable binding (§6.2), so we adopt a two-phase curriculum that parallels LLM pre-/post-training (Fig. 1). In the *pre-training* phase, the SAE is trained on reconstruction and sparsity, allowing the decoder to form a high-capacity dictionary. In subsequent *post-training*, we strengthen the binding and value losses, then activate the orthogonality penalty. This reshapes the latent space so supervised slots become clean, disentangled carriers of atomic concepts, while remaining decoupled from the free feature bank. This curriculum retains the benefits of joint optimization (concept slots that are simultaneously interpretable and task-predictive) while avoiding the degenerate minima that arise when strong supervision is applied before the underlying representation has stabilized (§C).

Objectives. We train the encoder, decoder, and value head jointly. Our binding objective augments the standard SAE loss ($\mathcal{L}_{\text{SAE}}$) with (i) a supervised loss that assigns each relation to a dedicated concept slot ($\mathcal{L}_{\text{align}}$), (ii) a z_{concept}–z_{rest} decorrelation penalty to reduce concept leakage into z_{rest} ($\mathcal{L}_{\perp}$), and (iii) an auxiliary value loss that encourages the concept slots to support answer prediction ($\mathcal{L}_{\text{val}}$), where y_{ans} is the ground-truth answer:

$$
\begin{aligned}
\mathcal{L}_{\text{SAE}} &= \lambda_{\text{rec}} \|h - \hat{h}\|_2^2 + \lambda_{\text{sp}} \|z\|_1, \\
\mathcal{L}_{\text{align}} &= \text{CE}\big(\text{softmax}(z_{\text{concept}}), y_{\text{rel}}\big), \\
\mathcal{L}_{\perp} &= \|\text{corr}(z_{\text{concept}}, z_{\text{rest}})\|_F^2, \\
\mathcal{L}_{\text{val}} &= \text{CE}\big(\text{softmax}(V(z_{\text{concept}})), y_{\text{ans}}\big), \\
\mathcal{L} &= \mathcal{L}_{\text{SAE}} + \lambda_{\text{align}} \mathcal{L}_{\text{align}} + \lambda_{\perp} \mathcal{L}_{\perp} + \lambda_{\text{val}} \mathcal{L}_{\text{val}}.
\end{aligned}
\tag{1}
$$

See §C.3 for full definitions and hyperparameters. We also assess each term through ablations in §6.

Overall, our method turns a frozen LLM's conceptual evidence into an explicit, verifiable, and controllable interface by (i) training an SAE on a single intermediate layer, (ii) allocating one-to-one ontology-aligned concept slots, and (iii) using a two-stage curriculum that yields disentangled, slot-addressable features. Because only the lightweight SAE is trained, the base model remains fixed, enabling efficient diagnosis (slot verification) and causal control (slot interventions).

[2]We choose GPT-2 for controlled analysis under compute limits, enabling layer-wise probing, ablation/steering analysis.

4 Implementation

We attach the SAE to a frozen base LLM (*e.g.*, GPT–2; our approach is not limited to a specific model) by extracting a representation $h \in \mathbb{R}^d$ from an intermediate layer ℓ. Concretely, given an input biography question x, the model produces token-level hidden states at each layer, and we pool the activations from layer ℓ (*e.g.*, mean pooling) to form h. The base LLM is kept frozen, preserving its linguistic competence while placing all supervision on the light-weight SAE interface (§B.4).

4.1 Concept-Aligned Sparse Autoencoder

We use a supervised SAE that exposes an interpretable control surface aligned with $\mathcal{R}$, while delegating residual variance to a large bank of free features. The encoder $E : \mathbb{R}^d \to \mathbb{R}^{|\mathcal{R}|+K}$ maps h to a sparse code $z = \mathrm{ReLU}(W_e h + b_e)$, partitioned as $z = [z_{\text{concept}}; z_{\text{rest}}]$ with $z_{\text{rest}} \in \mathbb{R}^K$ (unsupervised free features) and $z_{\text{concept}} \in \mathbb{R}^{|\mathcal{R}|}$ (supervised concept slots). The decoder $D : \mathbb{R}^{|\mathcal{R}|+K} \to \mathbb{R}^d$ reconstructs $\hat{h} = W_d z + b_d$. We set K=10,000 (1-hop) and K=100,000 (2-hop) to add capacity without burdening the $|\mathcal{R}|$ interpretable slots, read out concept evidence from z_{concept} at the output token, and post-train the SAE with the loss in §3.

4.2 Tasks and Datasets

We evaluate our approach AlignSAE on two controlled benchmarks designed to test both direct attribute retrieval and multi-step logical derivation: (i) *biography factual recall* (1-hop) and (ii) *multi-step compositional reasoning* (2-hop).

4.2.1 Factual Recall (1-hop)

We validate our approach on a biography QA task over a fixed ontology. Let $\mathcal{R}$ be a fixed ontology of $|\mathcal{R}| = 6$ (Table 6; §A). For a person p and relation $r \in \mathcal{R}$, a canonical table provides the gold value $y^\star = g(p,r)$ (*e.g.*, *Wesleyan University* for UNIVERSITY). Each input x is a natural-language question that mentions p and implicitly targets a single relation $r^\star \in \mathcal{R}$; the model must generate the corresponding value $y^\star = g(p,r^\star)$.

We adopt the synthetic biography dataset of Allen-Zhu & Li (2024) with minor modifications to better separate *semantic binding* from *template memorization*. We generate 1,000 person profiles (five biography variants each) and instantiate questions from paraphrase templates that preserve the same underlying relation. To assess robustness to surface form, we use a disjoint template split: we train on all profiles using two templates and evaluate Unseen-Template generalization on two held-out templates (Table 6; §A). The held-out templates use different lexical triggers (*e.g., born* vs. *birth city*, *alma mater* vs. *university*) to reduce n-gram overlap with training prompts. We also explored *LLM-generated questions* by few-shot prompting Claude 3.5 Sonnet on 1-hop RC queries (e_1,r) to generate instance-specific paraphrases, but excluded them since many are too indirect for base GPT-2 (details in Fig. 11; §A.4).

4.2.2 Compositional Reasoning (2-hop)

We further evaluate AlignSAE on a *two-hop* compositional reasoning task (Du et al., 2025). We use 2-hop as a compositional concept test for AlignSAE. Unlike 1-hop, it decomposes prediction into two concept steps with an intermediate entity e_2, so concept–slot binding can be *verified* and *intervened* at the step where each concept is actually used, rather than inferred only from final-answer accuracy. This decomposition also enables targeted causal tests (*e.g.*, swapping the r_2 at the second step), making controllability more diagnostic. Each example provides a start entity e_1 and relations (r_1,r_2); the model must infer $e_1 \xrightarrow{r_1} e_2 \xrightarrow{r_2} e_3$ (*i.e.*, e_2=$r_1(e_1)$, e_3=$r_2(e_2)$). Inputs use templates like *"Who is the r_2 of the r_1 of e_1?"* with brief profile sentences stating the two hop facts

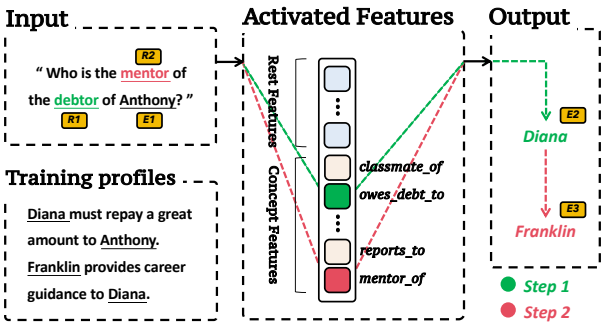


Figure 2: 2-hop step-wise alignment. With (e_1,r_1,r_2), the model predicts (e_2,e_3); r_1 activates at e_2 and r_2 at e_3, with remaining capacity in unsupervised features.

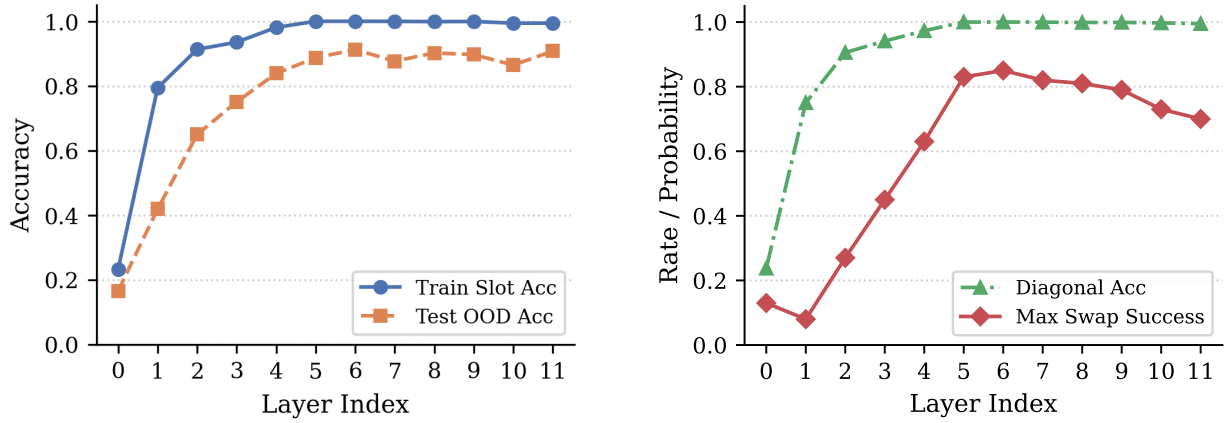


Figure 4: Binding generalization vs. causal control, showing performance on train and unseen distributions (left) alongside the corresponding diagonal and swap mechanisms (right).

(Fig. 14; §G). The dataset instantiates an ontology of 20 relations over 60 entities (see §G). Following Fig. 2, we train the model to output "$e_2; e_3$" and apply step-wise supervision: the r_1 slot should activate when generating e_2, and the r_2 slot when generating e_3.

4.3 Metrics

To measure concept–slot alignment, we build a relation–slot confusion matrix C where $C[r,j]$ is the normalized fraction of examples whose top-activated slot at the answer token is j for gold relation r, and report $\mathrm{Acc}_{\mathrm{bind}} := \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} C[r,r]$.

4.4 Controllability (Swap Test)

To evaluate *causal* controllability, we perform a *swap test*: given a query whose gold relation is $r^\star$, we intervene by injecting the decoded direction of an alternative relation slot $j \neq r^\star$ with strength $\alpha > 0$. Formally, letting $h_{\ell,t} \in \mathbb{R}^d$ denote the residual stream at layer ℓ and token position t, and $W_{\mathrm{dec}} \in \mathbb{R}^{d \times K}$ denote the SAE decoder with reconstruction $\hat{h}_{\ell,t} = W_{\mathrm{dec}} z_{\ell,t}$, the decoded basis direction for slot j is $v_j := W_{\mathrm{dec}} e_j$ and the intervention is $h'_{\ell,t} = h_{\ell,t} + \alpha v_j = h_{\ell,t} + W_{\mathrm{dec}}(\alpha e_j)$. We count a swap as successful if the model's top-1 generated answer matches the gold value under the intervened relation, *i.e.*, $g(p,j)$, rather than the original target $g(p,r^\star)$ (full definitions in §D).

5 Results: Factual Recall (1-hop)

We first evaluate the proposed interface across transformer layers, data (Train, Test–Unseen), and controllability settings. Metrics follow §4.3.

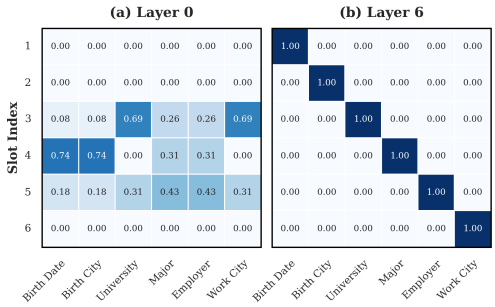


Figure 3: Concept–slot binding at a shallow layer (a) versus a mid layer (b). At layer 0, supervision for each relation is dispersed across multiple slots, whereas at layer 6 the SAE learns a perfect one-to-one binding.

5.1 Layer-Wise Performance

The contrast between layer 0 and layer 6 illustrates how concept alignment emerges (Fig. 3). At layer 0, the relation–slot confusion matrix is diffuse with substantial off-diagonal mass and overlapping activations, indicating entangled features that fail to isolate ontology relations. At layer 6, the same matrix collapses to a sharp diagonal with negligible off-diagonal mass, yielding a stable permutation that assigns each relation to a single slot.

Table 1 and Fig. 4 reveal a clear mid-layer sweet spot. Performance peaks around Layer 6, where binding is essentially one-to-one (*Diagonal Acc*= 1.00), swap controllability is highest, and paraphrase generalization remains strong (*Test Unseen Acc*= 0.912). In contrast, early layers are poorly aligned (Layer 0, *Diagonal Acc*= 0.238), with diffuse off-diagonal mass (Fig. 3). Although Layer 6 incurs higher reconstruction error, it yields a large gain in controllability (*swap success* +0.81), suggesting that clean semantic binding emerges in mid layers with richer contextual representations, while early embedding-dominated layers and deeper, more compressed layers make a stable slot interface harder to preserve.

Metric	Layer 0	Layer 6	Δ (L6 − L0)
Diagonal Acc ↑	0.238	**1.000**	↑ 0.76
Swap Success ↑	0.040	**0.847**	↑ 0.81
Train Slot Acc ↑	0.232	**1.000**	↑ 0.77
Test Unseen Acc ↑	0.165	**0.912**	↑ 0.75
Recon MSE ↓	$6.53 \cdot 10^{-5}$	$7.42 \cdot 10^{-2}$	↑ ≈ 1.1**k**×

Table 1: Layer 0 vs. Layer 6 performance comparison.

5.2 Concept–Feature Alignment

To quantify how cleanly each concept is represented at different depths, we compare a traditional SAE trained purely unsupervised with AlignSAE, which adds concept-supervision as a post-training signal. Let $z_i \in \mathbb{R}^{|\mathcal{R}|+K}$ be the sparse code for example i, and let $c(i)$ be its concept label (*e.g.*, one of six ontological relations). We first define the *average activation* of concept c on feature $k \in \{1, \ldots, |\mathcal{R}|\}$ as $A_{c,k} = \mathbb{E}_{i:c(i)=c}\left[z^{(i)}_{\text{concept},k}\right]$, and normalize over features to obtain a concept–feature distribution $B_{c,k} = \frac{A_{c,k}}{\sum_{k'} A_{c,k'} + \epsilon}$. From $B_{c,\cdot}$ we derive two summary metrics for each concept c:

Effective number of features (EffFeat). We measure how many features are effectively used to represent a concept via the entropy of $B_{c,\cdot}$: $\text{EffFeat}(c) = \exp\left(-\sum_k B_{c,k} \log B_{c,k}\right)$, where smaller values indicate that a concept is concentrated on fewer features (*i.e.*, lower fragmentation).

Top-1 concentration (Top1C). We also track how much of a concept's mass is captured by its single most responsive feature: $\text{Top1C}(c) = \max_k B_{c,k}$, where larger values indicate that one feature dominates the representation of concept c.

Fig. 5 reports layer-wise averages of these metrics over six concepts for AlignSAE and a traditional SAE. The traditional SAE (pre-training only) remains highly fragmented across all layers: EffFeat spans hundreds to thousands of features per concept and Top1C ≈ 0, suggesting that concepts are not consistently localized to a single feature. In contrast, AlignSAE sharply reduces fragmentation (EffFeat ≈ 1) and increases concentration via post-training supervision, yielding compact representations from mid layers onward and near one-to-one bindings in deeper layers. Overall, concept-level post-training converts a diffuse many-to-many feature space into a compact, interpretable interface with directly addressable concepts.

5.3 Swap Controllability

We probe whether concept slots behave as usable control knobs by measuring swap success under intervention (see §4.4). Fig. 6 reports average swap success rate (average over 1,000 examples) across layers and amplification strengths: moderate amplification ($\alpha \approx 2$) reliably switches the answer type at mid layers (5–8). At layer 6, success rises from 0.54 at α=1.0 to 0.85 at α=2.0, but drops to 0.23 at α=10.0, indicating that over-amplification destabilizes the intervention and revealing a narrow regime where control is effective and predictable. For example, for a BIRTH_DATE question , the model originally answers *"24, March, 1964"*; amplifying the UNIVERSITY slot with α=2 switches the output to *Wesleyan University*, showing that slots are causal control handles rather than merely diagnostic (more examples in Table 7 & Fig. 13; §F.2). We further ablate the post-training objectives; results in §6 show each component is necessary for robust controllability. Overall,

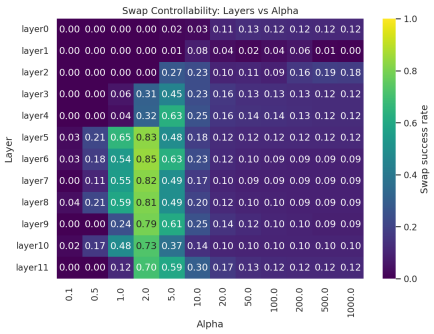
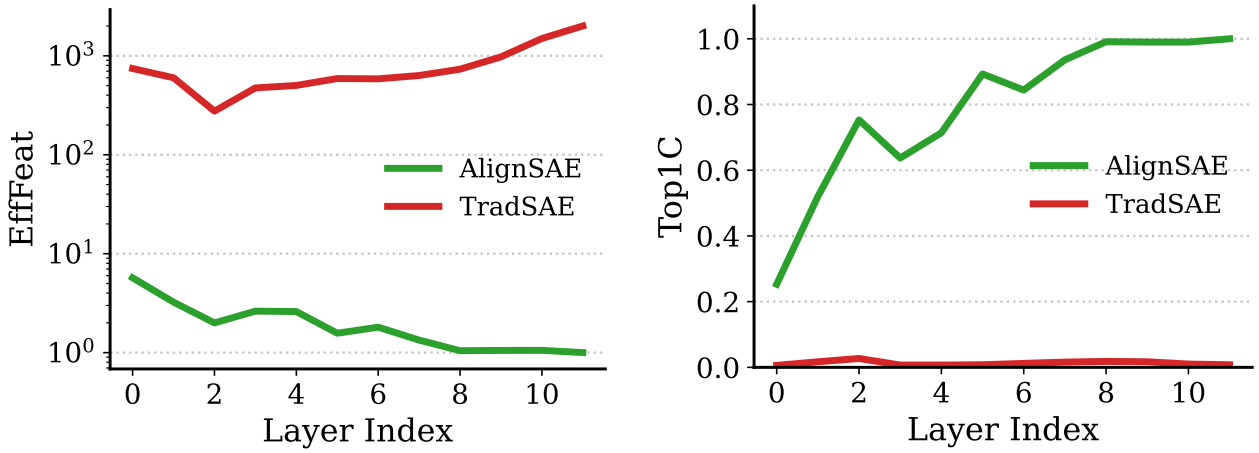


Figure 6: Swap controllability across layers and amplification α. Mid layers remain robust around $\alpha \approx 2$.



(a) Concept fragmentation (↓)

(b) Concept concentration (↑)

Figure 5: Layer-wise fragmentation (↓) and concentration (↑) for AlignSAE and a traditional SAE.

early layers are too local and deeper layers too compressed; mid-layer post-training (*e.g.*, Layer 6) yields near-perfect binding and reliable swaps at $\alpha \approx 2$, the practical regime for controllable concept access.

5.4 Swap Error Analysis

Swap	**Q:** Where did Jesse Kian Tate go to college? UNIV → **Swap:** MAJOR
Outputs	**Base:** Rochester Inst. of Tech. **Gold:** Physical Therapy **Gen:** **Geography** [Type ✔ Entity ✗]

Table 3: A failure case: steering ($\alpha = 10$) successfully flips the semantic category (type) but fails to retrieve the specific ground-truth entity.

Even when swap steering fails to hit the *exact* gold entity, it often preserves the *answer category* of the intended swapped concept. Table 2 reports *Category Preservation*, computed *only over failed swaps*. Under moderate amplification (α=2), 75.3% of failures remain in the correct category; under strong amplification (α=10), this rises to 83.0%, suggesting that larger interventions more reliably move the model onto the right target type even as exact entity selection degrades. For example, in Table 3, swapping UNIVERSITY→MAJOR at α=10 yields a ***type*** ✓ but ***entity*** ✗ answer: *Geography* instead of *Physical Therapy*. In this controlled setting, our strict swap metric may understate controllability: even without exact entity matches, interventions often enforce the target concept.

Target swap	α=2 **(15% Err)**			α=10 **(77% Err)**		
	S	**D**	**S%**	**S**	**D**	**S%**
birth_city	36	20	64.3	91	50	64.5
birth_date	20	0	100.0	139	0	100.0
employer	8	1	88.9	134	3	97.8
major	1	0	100.0	77	51	60.2
university	42	10	80.8	101	1	99.0
work_city	9	7	56.2	94	25	79.0
Overall	**116**	**38**	**75.3**	**636**	**130**	**83.0**

Table 2: Category retention on *failed* swaps at Layer 6. "Same" (S) means correct semantic class; "Diff" (D) is outside.

Config	Train Slot	Test-OOD Slot	Train Acc	Test-Unseen Acc	Swap Success Rate
joint	**1.000**	**0.912**	**1.000**	**0.809**	**0.847**
no_align	0.167	0.167	**1.000**	**0.809**	0.053
no_ortho	0.711	0.689	**1.000**	**0.809**	0.046
no_value	0.177	0.172	**1.000**	**0.809**	0.720

Table 4: **Ablations of AlignSAE post-training. Best** configuration is **joint**, which achieves strong OOD binding and high swap success simultaneously. Swap success is reported at each configuration's best-performing intervention strength (α: joint=2, no_align=0.1, no_ortho=10, no_value=100). Chance slot binding for 6 relations is $1/6 \approx 0.167$.

6 Ablation Study

6.1 Effect of Training Objectives

We ablate three post-training components of AlignSAE: (i) binding/alignment ($\mathcal{L}_{\text{align}}$), (ii) orthogonality ($\mathcal{L}_{\perp}$), and (iii) value/sufficiency ($\mathcal{L}_{\text{val}}$), while keeping the underlying task and backbone fixed.

Results. Binding is necessary for alignment: removing $\mathcal{L}_{\text{align}}$ collapses slot binding to chance (0.167 Test-Unseen) and nearly eliminates swap control (0.053). **Orthogonality is necessary for control:** without $\mathcal{L}_{\perp}$, binding remains high (0.689 Test-Unseen) but swaps fail (0.046), indicating leakage that breaks causal interventions. **Value improves interface quality:** removing $\mathcal{L}_{\text{val}}$ yields near-chance binding (0.172 Test-Unseen) yet can still achieve moderate swap success (0.72) at very large α, whereas the full model achieves both strong Test-Unseen binding (0.891) and high swap success (0.847) at a small $\alpha = 2$.

6.2 Effect of Stage-1 Pretraining

We ablate the unsupervised Stage 1 and train AlignSAE *from scratch* with the same post-training objective (*No Stage 1*), keeping all other settings fixed.

Results. As shown in Table 5, removing Stage 1 substantially worsens optimization and representation quality: total loss increases by 79.8% (1.86 → 3.34), with higher reconstruction error (L_{recon}: +61.1%) and a large degradation in sparsity (L_{sparse}: +1185.2%). Orthogonality also worsens (L_{ortho}: +41.8%). In contrast, slot binding accuracy changes little (slot_acc: +1.7%). Overall, Stage 1 is crucial for a stable sparse reconstructive basis, even if concept binding remains feasible without it.

Metric	No Stage 1	With Stage 1	Δ (With–No)	%Δ
$\mathcal{L}\downarrow$	3.3438	1.8594	$L\downarrow$ 1.4844	**-44.4%**
$L_{\text{recon}}\downarrow$	0.0412	0.0256	$L\downarrow$ 0.0156	**-37.9%**
$L_{\text{sparse}}\downarrow$	0.2682	0.0209	$L\downarrow$ 0.2473	**-92.2%**
$L_{\text{align}}\downarrow$	0.0605	0.0762	$L\uparrow$ 0.0157	**+26.0%**
$L_{\text{ortho}}\downarrow$	0.5330	0.3760	$L\downarrow$ 0.1570	**-29.5%**
$L_{\text{value}}\downarrow$	3.3169	3.4125	$L\uparrow$ 0.0956	**+2.9%**
Acc_{bind} (train)$\uparrow$	0.9700	1.0000	$A\uparrow$ 0.0300	**+3.1%**
$\text{Acc}_{\text{bind}}\uparrow$	0.9699	0.9538	$A\downarrow$ 0.0161	**-1.7%**
$L_{\text{indep}}\downarrow$	157.8025	4.7621	$L\downarrow$ 153.0404	**-97.0%**

Table 5: Stage-1 ablation. Deltas are computed as **With–No**; losses prefer ↓ and accuracies prefer ↑.

7 Results: Multi-Step Reasoning (2-hop)

Building on the layer sweep in Fig. 6, we use Layer 6 for all 2-hop experiments, as it provides the most reliable substrate for slot-level binding and causal control in the step-wise setting (§4.2.2).

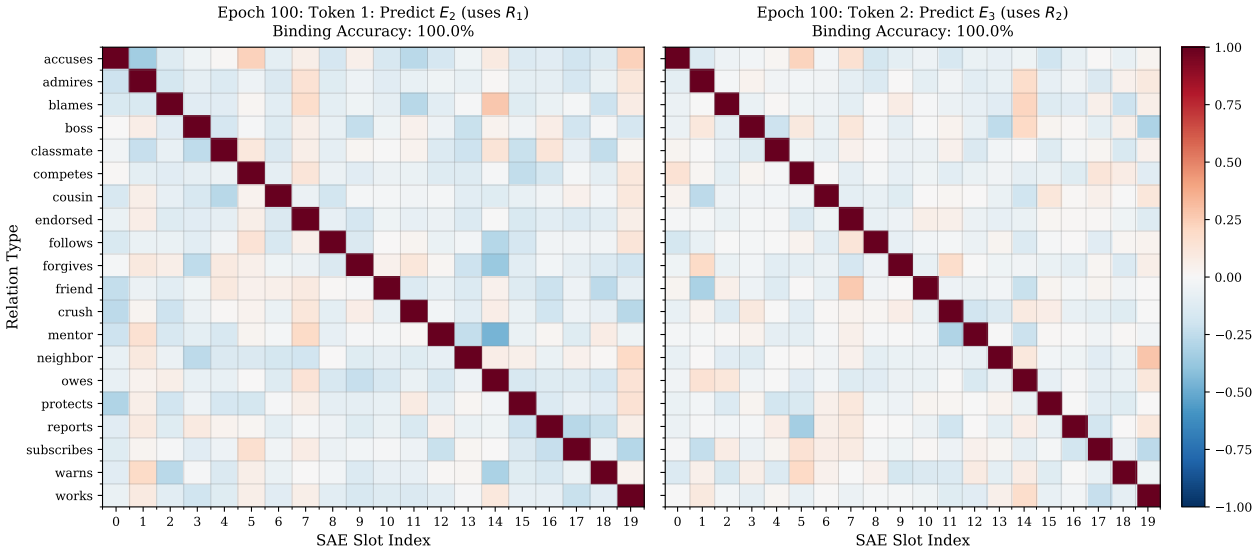


Figure 7: 2-hop Concept–slot binding. Confusion matrices at Step 1 (decoding e_2, binding to r_1) and Step 2 (decoding e_3, binding to r_2). The near-perfect diagonals indicate localization of concept to its designated slot.

7.1 Step-Wise Concept–Slot Binding

Fig. 7 shows that relation identity can be read from the designated concept slots at each decoding step. At the e_2 step (Token 1), the slot distribution concentrates on the slot assigned to r_1; at the e_3 step (Token 2), it concentrates on the slot assigned to r_2. Both confusion matrices exhibit sharp diagonals (100% binding accuracy), indicating clean step-wise alignment for 2-hop reasoning.

7.2 Swap Controllability

We next evaluate if the aligned slots are causal control knobs. Following the intervention protocol in §4.4, we inject a decoded direction corresponding to a target relation slot with strength α during inference and measure whether the model's output switches to the entity implied by the swapped relation. We compare against a traditional SAE and a supervised linear probe (Rimsky et al., 2024). We train a linear probe to predict the relation label from the frozen hidden state using cross-entropy loss ($\sim 100\%$ concept-classification accuracy). For swapping, we treat each classifier row weight w_r as a concept direction and intervene directly in hidden space via $h'_{\ell,t} = h_{\ell,t} - \alpha w_{r_{\text{input}}} + \alpha w_{r_{\text{target}}}$. As shown in Fig. 8, the supervised AlignSAE enables higher swap success than both the traditional SAE and linear probe, peaking near $\alpha \approx 50$.

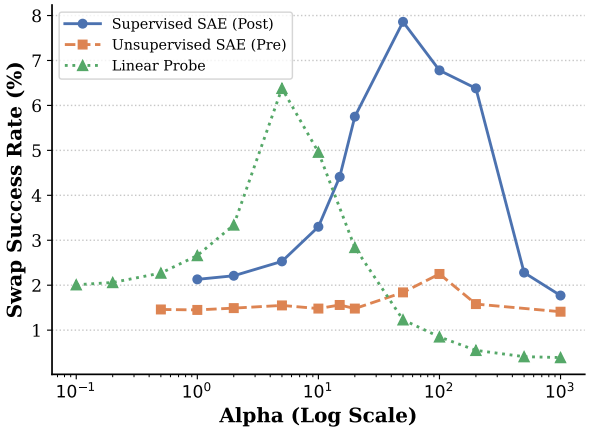


Figure 8: Swap controllability in 2-hop reasoning. The post-trained AlignSAE achieves 4× higher swap success than the traditional SAE when $\alpha \approx 50$.

7.3 Understanding Grokking via Compositional Concept Binding

We first observe grokking-like training dynamics in the *base* two-hop model: loss decreases smoothly, while validation accuracy remains low for many epochs before a sharp transition (Fig. 15). We then use AlignSAE as a diagnostic probe to characterize *what changes internally* across this transition. AlignSAE is well-suited for this analysis because it provides (i) *localization*: it encourages the model to consolidate evidence for a human-defined concept into a single, addressable slot instead of spreading it across many latent features, and (ii) *verification*: once concepts are slot-addressable, we can directly inspect which concept the model

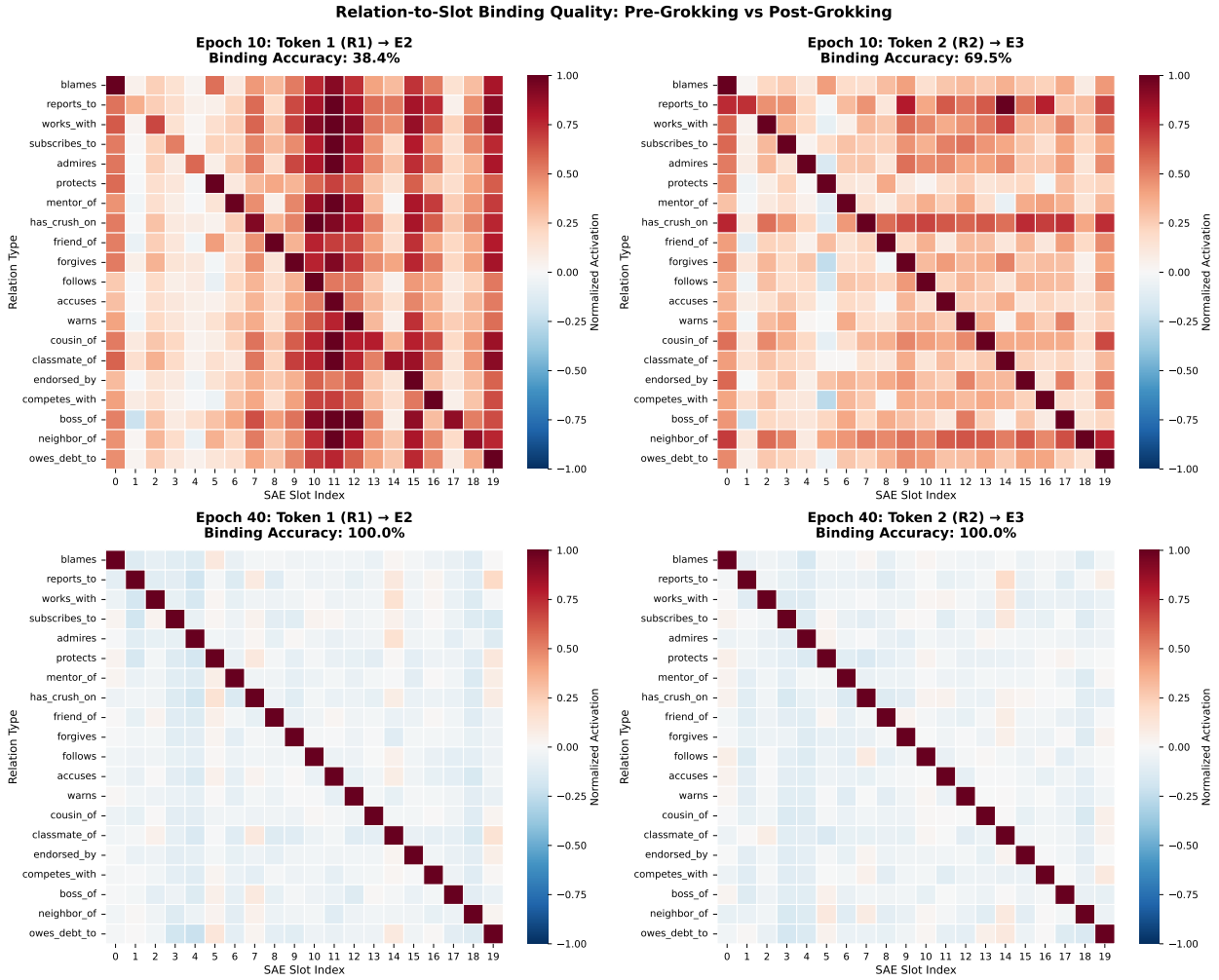


Figure 9: **Concept binding (pre-grokking vs. post-grokking).** We visualize concept–slot confusion matrices at e_2 (Token 1, supervised to R_1) and e_3 (Token 2, supervised to R_2). Binding is initially entangled but becomes perfectly diagonal after the grokking transition, indicating clean step-wise alignment.

is effectively "using" at each step by reading out the active slot at the answer token.

Our working hypothesis is that *before grokking*, the model can fit the training set in a "messy" regime where step-specific concept evidence is present but not cleanly separated: the intended concept for a given step is encoded as a partially entangled mixture over slots. In this phase, the model may appear to "know" the task, yet it has not formed a stable compositional interface that isolates the intermediate-step concept from the final-step concept. After the grokking transition, we expect a qualitative re-organization: internal representations re-organize into structured, compositional concept features, yielding stable step-wise 1-to-1 binding (Token 1 activates the concept for step 1; Token 2 activates the concept for step 2).

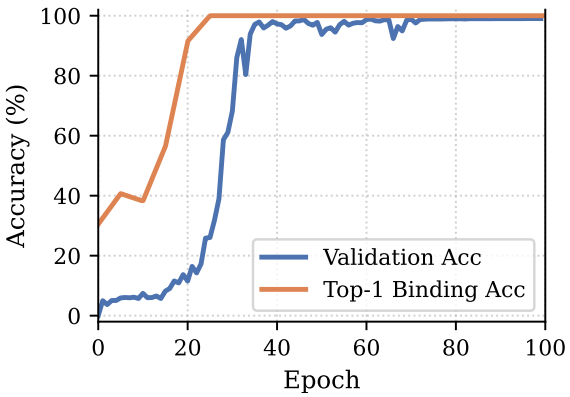


Figure 10: **Grokking-like emergence.** Top-1 binding accuracy rises sharply and saturates earlier, while validation accuracy improves later, consistent with a delayed generalization transition in 2-hop reasoning.

This perspective predicts a *lag between knowing and showing*. In Fig. 10, Top-1 binding accuracy climbs rapidly and reaches a near-ceiling level early in training, whereas validation accuracy lags behind and improves later with a delayed jump. This dissociation suggests a phase of "hidden" progress: the model first organizes the underlying concept structure into stable internal features ("knowing"), and only subsequently learns to consistently *use* that structure to produce correct out-of-distribution generalization ("showing"). An intuitive analogy is organizing a disordered library: early effort assigns genres to fixed shelves (the internal system forms), but it takes additional practice before one can reliably retrieve the right book quickly from that system (the external performance jump).

Fig. 9 visualizes the representational transition at Token 1 (the e_2 step). Pre-grokking, binding remains diffuse (epoch 10: 38.4%), with substantial off-diagonal mass indicating that the step-1 concept is not yet isolated into its aligned slot. Post-grokking, the confusion matrix becomes perfectly diagonal (epoch 40: 100%), indicating that the model has consolidated the step-1 concept into a dedicated slot. Together with the delayed jump in validation accuracy, these results are consistent with grokking as a phase transition from entangled concept representations to a clean, compositional regime in which step-specific concepts become distinct, slot-addressable features that support multi-step generalization.

8 Conclusion and Future Work

In this work, we present AlignSAE, a framework that upgrades Sparse Autoencoders (SAEs) from descriptive probes to controllable interfaces. To mitigate feature entanglement in unsupervised SAEs, we introduce a "pre-train, then post-train" curriculum that enforces explicit alignment between designated latents and an external ontology, while retaining a large residual bank for reconstruction. This design yields reliable causal control, validated via paraphrase, robust concept swaps, and transfers cleanly to multi-hop reasoning. AlignSAE also enables fine-grained analysis of representation learning dynamics. We find a mechanistic account of grokking in compositional tasks: as generalization emerges, diffuse evidence concentrates into stable, step-wise bindings between relations and aligned slots. Furthermore, by grounding abstract concepts in latent structure, AlignSAE moves SAE-based interpretability toward controllable internal representations.

Looking forward, several promising directions remain for extending AlignSAE. A primary objective is scaling the alignment curriculum to broader, open-ended ontologies and investigating "hierarchical binding," where latent slots represent categories that can be further decomposed into granular sub-features. Additionally, we envision AlignSAE serving as a real-time "fact-check" layer, by monitoring the activation of factual concept slots during inference, one could detect and mitigate hallucinations before they reach the output. Finally, extending this framework to multimodal architectures would allow for a cross-modal interface to audit and steer how models reason. By evolving these latent structures from static probes into dynamic components, we aim to move toward AI systems that are inherently safer, more interpretable, and surgically editable.

Limitations

AlignSAE is evaluated in a deliberately controlled setting on a frozen GPT-2, using synthetic 1-hop biography QA and a 2-hop reasoning task rendered into natural-language prompts (§5, §7). This setup makes concept binding and causal interventions precisely measurable (§5.3) and allows us to extend grokking-style mechanistic analysis to the *concept-binding* level by tracking binding quality across training (Fig. 10 and Fig. 9). Unlike prior work on grokked transformers (Wang et al., 2024; Ye et al., 2025), which train on explicit symbolic IDs, we intentionally avoid entity/relation identifiers and instead use natural-language templates to better mimic realistic usage and reduce shortcut learning; this choice improves validity but also makes generalization harder and leaves open how our findings translate to ID-based settings. More broadly, our conclusions are currently limited to a GPT-2 backbone and templated data with paraphrase splits (§A.4); highly indirect LLM-generated question styles can exceed the backbone's QA capacity (Unseen-Template generalization; §A.4), so we focus on unseen but direct template-level questions to ensure failures reflect representation-level alignment rather than backbone incapability. We also use a small ontology (*e.g.,* 20 relations in 2-hop), which does not reflect full-scale SAE deployments; scaling to larger ontologies/models is left to future work, though the proposed binding and causal evaluation are not tied to this size.

References

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of Language Models: Part 3.1, Knowledge Storage and Extraction. In *Proceedings of the 41st International Conference on Machine Learning*, ICML '24, July 2024. Full version available at https://ssrn.com/abstract=5250633.

Dana Arad, Aaron Mueller, and Yonatan Belinkov. Saes are good for steering – if you select the right features, 2025. URL https://arxiv.org/abs/2505.20063.

Reza Bayat, Ali Rahimi-Kalahroudi, Mohammad Pezeshki, Sarath Chandar, and Pascal Vincent. Steering large language model activations in sparse spaces, 2025. URL https://arxiv.org/abs/2503.00177.

Leonard Bereska and Stratis Gavves. Mechanistic interpretability for AI safety - a review. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=ePUVetPKu6. Survey Certification, Expert Certification.

Amrita Bhattacharjee, Shaona Ghosh, Traian Rebedea, and Christopher Parisien. Towards inference-time category-wise safety steering for large language models, 2024. URL https://arxiv.org/abs/2410.01174.

Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. Language models can explain neurons in language models. https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html, 2023.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Sviatoslav Chalnev, Matthew Siu, and Arthur Conmy. Improving steering vectors by targeting sparse autoencoder features, 2024. URL https://arxiv.org/abs/2411.02193.

David Chanin, Tomáš Dulka, and Adrià Garriga-Alonso. Feature hedging: Correlated features break narrow sparse autoencoders, 2025. URL https://arxiv.org/abs/2505.11756.

Jingyi Cui, Qi Zhang, Yifei Wang, and Yisen Wang. On the theoretical understanding of identifiable sparse autoencoders and beyond, 2025. URL https://arxiv.org/abs/2506.15963.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models, 2023. URL https://arxiv.org/abs/2309.08600.

Dina Demner-Fushman, Sophia Ananiadou, Makoto Miwa, and Junichi Tsujii (eds.). *Proceedings of the 24th Workshop on Biomedical Language Processing*, Viena, Austria, August 2025. Association for Computational Linguistics. ISBN 979-8-89176-275-6. doi: 10.18653/v1/2025.bionlp-1.0. URL https://aclanthology.org/2025.bionlp-1.0/.

Yupei Du, Philipp Mondorf, Silvia Casola, Yuekun Yao, Robert Litschko, and Barbara Plank. Reason to rote: Rethinking memorization in reasoning. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 8670–8690, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.437. URL https://aclanthology.org/2025.emnlp-main.437/.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. *Transformer Circuits Thread*, 2022. https://transformer-circuits.pub/2022/toy_model/index.html.

Eoin Farrell, Yeu-Tong Lau, and Arthur Conmy. Applying sparse autoencoders to unlearn knowledge in language models, 2024. URL https://arxiv.org/abs/2410.19278.

Javier Ferrando, Gabriele Sarti, Arianna Bisazza, and Marta R. Costa-jussà. A primer on the inner workings of transformer-based language models, 2024. URL https://arxiv.org/abs/2405.00208.

Marc Fischer, Mislav Balunovic, Dana Drachsler-Cohen, Timon Gehr, Ce Zhang, and Martin Vechev. Dl2: training and querying neural networks with logic. In *International Conference on Machine Learning*, pp. 1931–1941. PMLR, 2019.

Kuzman Ganchev, João Graça, Jennifer Gillenwater, and Ben Taskar. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*, 11(67):2001–2049, 2010. URL http://jmlr.org/papers/v11/ganchev10a.html.

Shaona Ghosh, Amrita Bhattacharjee, Yftah Ziser, and Christopher Parisien. Safesteer: Interpretable safety steering with refusal-evasion in llms, 2025. URL https://arxiv.org/abs/2506.04250.

Ralph Grishman and Beth Sundheim. Message Understanding Conference- 6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*, 1996. URL https://aclanthology.org/C96-1079/.

Phillip Guo, Aaquib Syed, Abhay Sheshadri, Aidan Ewart, and Gintare Karolina Dziugaite. Mechanistic unlearning: Robust knowledge unlearning and editing via mechanistic localization, 2024. URL https://arxiv.org/abs/2410.12949.

Ruben Härle, Felix Friedrich, Manuel Brack, Björn Deiseroth, Stephan Waeldchen, Patrick Schramowski, and Kristian Kersting. Measuring and guiding monosemanticity. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=REHjkmWdQL.

Zhengfu He, Wentao Shu, Xuyang Ge, Lingjie Chen, Junxuan Wang, Yunhua Zhou, Frances Liu, Qipeng Guo, Xuanjing Huang, Zuxuan Wu, Yu-Gang Jiang, and Xipeng Qiu. Llama scope: Extracting millions of features from llama-3.1-8b with sparse autoencoders, 2024. URL https://arxiv.org/abs/2410.20526.

Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. Harnessing deep neural networks with logic rules. In Katrin Erk and Noah A. Smith (eds.), *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2410–2420, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1228. URL https://aclanthology.org/P16-1228/.

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=F76bwRSLeK.

Yi Jing, Zijun Yao, Hongzhu Guo, Lingxu Ran, Xiaozhi Wang, Lei Hou, and Juanzi Li. Lingualens: Towards interpreting linguistic mechanisms of large language models via sparse auto-encoder, 2025. URL https://arxiv.org/abs/2502.20344.

Adam Karvonen, Can Rager, Johnny Lin, Curt Tigges, Joseph Bloom, David Chanin, Yeu-Tong Lau, Eoin Farrell, Callum McDougall, Kola Ayonrinde, Matthew Wearden, Arthur Conmy, Samuel Marks, and Neel Nanda. Saebench: A comprehensive benchmark for sparse autoencoders in language model interpretability, 2025. URL https://arxiv.org/abs/2503.09532.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models, 2020. URL https://arxiv.org/abs/2007.04612.

Patrick Leask, Bart Bussmann, Michael T Pearce, Joseph Isaac Bloom, Curt Tigges, Noura Al Moubayed, Lee Sharkey, and Neel Nanda. Sparse autoencoders do not find canonical units of analysis. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=9ca9eHNrdH.

Sihang Li, Wei Shi, Ziyuan Xie, Tao Liang, Guojun Ma, and Xiang Wang. Safer: Probing safety in reward models with sparse autoencoder, 2025. URL https://arxiv.org/abs/2507.00665.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

Junru Lu, Gabriele Pergola, Lin Gui, and Yulan He. Event knowledge incorporation with posterior regularization for event-centric question answering, 2023. URL https://arxiv.org/abs/2305.04522.

Aleksandar Makelov, Georg Lange, and Neel Nanda. Saes discover meaningful features in the ioi task. Alignment Forum, 2024. URL https://www.alignmentforum.org/posts/zj3GKWAnhPgTARByB/saes-discover-meaningful-features-in-the-ioi-task.

Samuel Marks, Can Rager, Eric J. Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models, 2025. URL https://arxiv.org/abs/2403.19647.

Gouki Minegishi, Hiroki Furuta, Yusuke Iwasawa, and Yutaka Matsuo. Rethinking evaluation of sparse autoencoders through the representation of polysemous words. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=HpUs2EXjOl.

Aashiq Muhamed, Mona T. Diab, and Virginia Smith. Decoding dark matter: Specialized sparse autoencoders for interpreting rare concepts in foundation models. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 1604–1635, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.87. URL https://aclanthology.org/2025.findings-naacl.87/.

Anh Nguyen, Jason Yosinski, and Jeff Clune. Understanding neural networks via feature visualization: A survey, 2019. URL https://arxiv.org/abs/1904.08939.

Kyle O'Brien, David Majercak, Xavier Fernandes, Richard Edgar, Blake Bullwinkel, Jingya Chen, Harsha Nori, Dean Carignan, Eric Horvitz, and Forough Poursabzi-Sangdeh. Steering language model refusal with sparse autoencoders, 2025. URL https://arxiv.org/abs/2411.11296.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. URL https://arxiv.org/abs/2203.02155.

Senthooran Rajamanoharan, Tom Lieberum, Nicolas Sonnerat, Arthur Conmy, Vikrant Varma, János Kramár, and Neel Nanda. Jumping ahead: Improving reconstruction fidelity with jumprelu sparse autoencoders, 2024. URL https://arxiv.org/abs/2407.14435.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering llama 2 via contrastive activation addition. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.828. URL https://aclanthology.org/2024.acl-long.828/.

Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeff Wu, Lucius Bushnaq, Nicholas Goldowsky-Dill, Stefan Heimersheim, Alejandro Ortega, Joseph Bloom, Stella Biderman, Adria Garriga-Alonso, Arthur Conmy, Neel Nanda, Jessica Rumbelow, Martin Wattenberg, Nandi Schoots, Joseph Miller, Eric J. Michaud, Stephen Casper, Max Tegmark, William Saunders, David Bau, Eric Todd, Atticus Geiger, Mor Geva, Jesse Hoogland, Daniel Murfet, and Tom McGrath. Open problems in mechanistic interpretability, 2025. URL https://arxiv.org/abs/2501.16496.

Dong Shu, Xuansheng Wu, Haiyan Zhao, Daking Rai, Ziyu Yao, Ninghao Liu, and Mengnan Du. A survey on sparse autoencoders: Interpreting the internal mechanisms of large language models. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 1690–1712, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.89. URL https://aclanthology.org/2025.findings-emnlp.89/.

Boshi Wang, Xiang Yue, Yu Su, and Huan Sun. Grokked transformers are implicit reasoners: A mechanistic journey to the edge of generalization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://arxiv.org/pdf/2405.15071.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=gEZrGCozdqR.

Xinyuan Yan, Shusen Liu, Kowshik Thopalli, and Bei Wang. Visual exploration of feature relationships in sparse autoencoders with curated concepts, 2025. URL https://arxiv.org/abs/2511.06048.

Jingyuan Yang, Rongjun Li, Weixuan Wang, Ziyu Zhou, Zhiyong Feng, and Wei Peng. Lf-steering: Latent feature activation steering for enhancing semantic consistency in large language models, 2025. URL https://arxiv.org/abs/2501.11036.

Zhun Yang, Joohyung Lee, and Chiyoun Park. Injecting logical constraints into neural networks via straight-through estimators, 2023. URL https://arxiv.org/abs/2307.04347.

Jiaran Ye, Zijun Yao, Zhidian Huang, Liangming Pan, Jinxin Liu, Yushi Bai, Amy Xin, Liu Weichuan, Xiaoyin Che, Lei Hou, and Juanzi Li. How do transformers learn implicit reasoning? In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=19ygs48nOa.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. Instruction tuning for large language models: A survey, 2025. URL https://arxiv.org/abs/2308.10792.

Yu Zhao, Alessio Devoto, Giwon Hong, Xiaotang Du, Aryo Pradipta Gema, Hongru Wang, Xuanli He, Kam-Fai Wong, and Pasquale Minervini. Steering knowledge selection behaviours in LLMs via SAE-based representation engineering. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5117–5136, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.264. URL https://aclanthology.org/2025.naacl-long.264/.

Mantong Zhou, Minlie Huang, and Xiaoyan Zhu. Robust reading comprehension with linguistic constraints via posterior regularization, 2019. URL https://arxiv.org/abs/1911.06948.

Content of Appendix

A Biography Dataset Generation
B Base Language Model Training
C Supervised Sparse Autoencoder
D Evaluation Metrics
E SAE Feature Activations by Relation Type
F Additional Qualitative Swap Examples
G Two Hop Reasoning Dataset Construction

A Biography Dataset Generation

A.1 Synthetic Biography Dataset

We generated 1,000 synthetic person profiles, each containing six factual attributes: birth date, birth city, university, major, employer, and work city. Each person was paired with 5 biography variants constructed from template-based generation. For question-answering evaluation, we employed a template-split strategy: templates 0–1 were designated for training (in-distribution), while templates 2–3 served as out-of-distribution test cases to evaluate semantic generalization beyond pattern matching.

A.2 Entity Vocabulary

The dataset drew from the following entity sets:

- **First names:** 411 diverse given names spanning traditional and modern choices
- **Middle names:** 461 names used for middle name generation
- **Last names:** 1,002 surnames representing common American family names
- **Birth cities:** 50 major U.S. cities (*e.g.*, New York, Los Angeles, Chicago, etc.)
- **Universities:** 341 U.S. colleges and universities spanning liberal arts colleges, research universities, technical institutes, and military academies
- **Majors:** 101 academic fields ranging from STEM disciplines (Computer Science, Mechanical Engineering, Biochemistry) to humanities (Philosophy, Art History, Creative Writing) and professional programs (Business Administration, Nursing, Architecture)
- **Companies:** 327 major U.S. corporations with associated headquarters cities, covering diverse industries including technology, finance, healthcare, retail, and manufacturing

This vocabulary size enables the generation of approximately $411 \times 461 \times 1{,}002 \times 50 \times 341 \times 101 \times 327 \approx 1.04 \times 10^{17}$ unique person profiles, ensuring minimal memorization pressure and focusing evaluation on semantic understanding rather than rote learning. We pick 1000 samples (6000 entities) in our dataset for the factual knowledge memorization.

A.3 Question-Answer Templates

Each of the six semantic relations was probed using four distinct question templates, enabling controlled evaluation of template generalization. Table 6 lists all templates used in our experiments.

This template design ensures semantic diversity while maintaining consistent information content, allowing us to test whether the SAE captures abstract semantic relations rather than surface-level linguistic patterns.

A.4 Using an LLM to generate unseen questions

Beyond the fixed template split (T0–T3; Table 6), we explored generating additional *unseen* questions with an external LLM. Concretely, for each 1-hop RC query (e_1,r) (entity e_1 and relation $r \in \mathcal{R}$), we few-shot prompt Claude 3.5 Sonnet via the OpenRouter API with two instantiated in-distribution questions (from T0–T1 for the same relation) and ask it to produce exactly two new, *simple and direct* paraphrases that

Relation	Train templates (T0–T1)	Held-out templates (T2–T3)
Birth Date	When was {FULL_NAME} born? On what date was {FULL_NAME} born?	What is {FULL_NAME}'s birth date? Can you tell me the birth date of {FULL_NAME}?
Birth City	Where was {FULL_NAME} born? In what city was {FULL_NAME} born?	What is {FULL_NAME}'s birth city? Can you tell me the birth city of {FULL_NAME}?
University	Where did {FULL_NAME} go to college? Which college did {FULL_NAME} attend?	What is {FULL_NAME}'s alma mater? Which university did {FULL_NAME} attend?
Major	What was {FULL_NAME}'s major? What is {FULL_NAME}'s field of study?	What did {FULL_NAME} study? What field did {FULL_NAME} study in?
Employer	Who does {FULL_NAME} work for? What company does {FULL_NAME} work for?	What is {FULL_NAME}'s employer? Which company employs {FULL_NAME}?
Work City	Where does {FULL_NAME} work? What city does {FULL_NAME} work in?	What is {FULL_NAME}'s work city? In which city is {FULL_NAME} employed?

Table 6: Template split for surface-form robustness. T0–T1 are used for training, while T2–T3 are held out for UNSEEN-TEMPLATE evaluation; the split is designed to reduce lexical overlap in relation triggers (*e.g.*, *born* vs. *birth date*, *work for* vs. *employer*).

query the same attribute. We enforce basic constraints (use the exact entity name, keep questions short, return exactly two lines); the full prompt is shown in Fig. 11. This procedure produces *instance-specific* unseen questions (unique per (e_1,r)), rather than global templates shared across entities.

In practice, many generations drift toward more indirect or complex phrasings that the base GPT-2 backbone cannot reliably answer, even when the questions are semantically well-formed. To isolate the effect of concept alignment (rather than backbone capacity), we therefore restrict the main experiments to the controlled template split in Table 6 and treat LLM-generated unseen questions as an exploratory stress test.

Examples. Compared to the fixed templates, Claude 3.5 Sonnet often introduces indirect phrasing. For BIRTH_DATE: (i) *"Do you happen to know the calendar date that marks Todd Raul Hanson's arrival?"* and (ii) *"I'm trying to find out the calendar date on which Edith Rocky Taylor made her first appearance."* (answer: `12,August,1991`). For EMPLOYER: *"I'm curious about the business where Jonathan Kiera Carney earns their living – do you know it?"* (answer: `Caterpillar Inc.`).

B Base Language Model Training

B.1 Model Architecture

We employed GPT-2 with 124M parameters as the base causal language model, featuring 768-dimensional hidden representations and 12 transformer layers.

B.2 Training Objective

The model was trained using a two-component curriculum: (1) *biography memorization*, where the model learned to predict entire biography sequences, and (2) *pure question-answering*, where only answer tokens contributed to the loss while question prompts were masked (label = -100).

User prompt used for OOD question generation (verbatim, with placeholders).

```
You are helping generate simple question variations for a dataset. Your task is to create 2 NEW questions that ask
about {RELATION_DESC} for the person "{FULL_NAME}".

IMPORTANT REQUIREMENTS:
1. Keep questions SIMPLE and DIRECT - similar to these training examples but with SLIGHT variations:
   - "{ID_EXAMPLE_1}"
   - "{ID_EXAMPLE_2}"

2. Make SMALL changes only - change just 1-2 words or reorder slightly
3. Questions must be answerable with a direct factual answer (date, city, university name, etc.)
4. DO NOT ask questions that require explanation or opinion
5. Use the exact name "{FULL_NAME}" in each question
6. Keep questions SHORT (under 15 words)

Generate exactly 2 simple question variations, one per line. Do NOT number them or add any other text.
```

Figure 11: Few-shot prompt used to generate OOD paraphrases for each (p,r) pair. {RELATION_DESC} is a short natural-language description of the target relation (*e.g., "the person's birth date"*), and {ID_EXAMPLE_1}–{ID_EXAMPLE_2} are instantiated from the ID training templates (T0–T1).

B.3 Optimization Hyperparameters

Selection protocol. We did not perform an extensive hyperparameter search for base LM training. Unless otherwise noted, we mainly adopt the optimization settings reported in Allen-Zhu & Li (2024) (AdamW with warmup, cosine decay, and gradient clipping) and keep them fixed across all runs. We only ran minimal sanity checks to ensure training stability (no divergence) and to achieve near-saturated train/validation performance on our synthetic curriculum; we did not tune hyperparameters to optimize downstream controllability or binding metrics.

Training was conducted with the following configuration:

- **Maximum training steps:** 80,000
- **Effective batch size:** 96 (distributed across available GPUs)
- **Learning rate schedule:** Linear warmup to 1×10^{-3} over 1,000 steps, followed by cosine annealing to 1×10^{-4}
- **Optimizer:** AdamW with weight decay 0.1 and $\epsilon = 1 \times 10^{-6}$
- **Gradient clipping:** Maximum norm 1.0
- **Maximum sequence length:** 512 tokens
- **Checkpoint frequency:** Every 10,000 steps

B.4 Activation Collection

Hidden states were extracted from the residual stream at the final token position of the question prompt (immediately before answer generation), representing the point where the model "decides" what information to retrieve. We collected activations from all 12 transformer layers independently to analyze the emergence of semantic binding across network depth.

C Concept-Aligned Sparse Autoencoder

C.1 Model Architecture

Our supervised SAE extends a standard sparse autoencoder with $|\mathcal{R}|$ dedicated relation slots, while delegating residual variance to a large bank of free features. The configuration is:

- **Input dimension:** d=768 (GPT-2 hidden size)
- **Total latent features:** $K+|\mathcal{R}|$ with $K \in \{10000, 100000\}$ (task-dependent) and $|\mathcal{R}|$
  - *Free features:* $z_{\text{rest}} \in \mathbb{R}^K$ (unsupervised)

  - *Relation slots:* $z_{\text{concept}} \in \mathbb{R}^{|\mathcal{R}|}$ (supervised; one per relation)
- **Encoder:** $E : \mathbb{R}^d \to \mathbb{R}^{|\mathcal{R}|+K}, \quad z = \text{ReLU}(W_e h + b_e)$ with $z = [z_{\text{concept}}; z_{\text{rest}}]$
- **Decoder:** $D : \mathbb{R}^{|\mathcal{R}|+K} \to \mathbb{R}^d, \ \hat{h} = W_d z + b_d$

Binding Mechanism. We partition $z = [z_{\text{concept}}; z_{\text{rest}}]$ with $z_{\text{concept}} \in \mathbb{R}^{|\mathcal{R}|}$ and define $p(r \mid x) = \text{softmax}(z_{\text{concept}})$. The binding loss applies cross-entropy with the one-hot gold relation, encouraging the correct relation slot to dominate.

C.2 Multi-Stage Training Protocol

Training proceeds in two stages to ensure stable convergence:

- **Stage 1 (Reconstruction-Only):** 100 epochs focusing solely on autoencoding quality before introducing binding constraints
- **Stage 2 (Full Supervision):** 500 epochs with complete loss function

C.3 Loss Function

The total training objective (see Section 4 of the main paper) combines six components. In Stage 1 (100 epochs), only reconstruction loss is active to stabilize the encoder-decoder. In Stage 2 (500 epochs), all six losses are jointly optimized:

$$\begin{aligned}\mathcal{L}_{\text{total}} = \lambda_{\text{recon}}\mathcal{L}_{\text{recon}} + \lambda_{\text{sparse}}\mathcal{L}_{\text{sparse}} + \lambda_{\text{align}}\mathcal{L}_{\text{align}} \\ + \lambda_{\text{ortho}}\mathcal{L}_{\text{ortho}} + \lambda_{\text{value}}\mathcal{L}_{\text{value}}\end{aligned} \tag{2}$$

Each component serves a specific purpose:

Reconstruction Loss.

$$\mathcal{L}_{\text{recon}} = \text{MSE}(\hat{h}, h) = \frac{1}{d}\sum_{i=1}^{d}(\hat{h}_i - h_i)^2 \tag{3}$$

Measures mean squared error between original activation h and reconstructed activation $\hat{h} = W_{\text{dec}} \cdot z$, where $d = 768$ is the hidden dimension. This ensures the SAE preserves information necessary for the language model's downstream predictions while learning a compressed latent representation.

Sparsity Loss.

$$\mathcal{L}_{\text{sparse}} = \frac{1}{B \cdot n_{\text{free}}}\sum_{b=1}^{B}\sum_{j=1}^{n_{\text{free}}}|z_{b,j}| \tag{4}$$

Enforces L1 penalty on the large number of free slots across batch size B, encouraging the model to activate only a small subset of features per sample. Sparse activations improve interpretability by ensuring each latent feature captures distinct semantic properties rather than distributing information diffusely.

Alignment Loss.

$$\mathcal{L}_{\text{align}} = \text{CrossEntropy}(\text{softmax}(z_{\text{rel}}), y) \tag{5}$$

$$= -\frac{1}{B}\sum_{b=1}^{B}\sum_{r=1}^{6} y_{b,r} \log \frac{\exp\big(z_{b,n_{\text{free}}+r}\big)}{\sum_{r'=1}^{6}\exp\big(z_{b,n_{\text{free}}+r'}\big)}. \tag{6}$$

Provides supervised guidance where y is a one-hot vector with $y_{b,\text{rule_idx}_b} = 1$ indicating the ground-truth relation type. This cross-entropy loss over softmax-normalized relation slot activations enforces that the 6 relation slots (indices $n_{\text{free}} + 1$ through $n_{\text{free}} + 6$) form a probability distribution with mass concentrated on the correct semantic relation. This is the binding loss used in all main experiments, enabling explicit classification of question types to specific latent dimensions.

Orthogonality Loss.

$$\mathcal{L}_{\text{ortho}} = \sum_{r=1}^{6} \sum_{j=1}^{n_{\text{free}}} \left(\frac{1}{B} \sum_{b=1}^{B} (z_{b,n_{\text{free}}+r} - \bar{z}_r)(z_{b,j} - \bar{z}_j) \right)^2 \tag{7}$$

Enforces statistical independence between supervised relation slots and unsupervised free slots by minimizing their cross-covariance. This prevents relation slots from encoding information already captured by free features, ensuring clean separation between task-specific and general-purpose representations.

Value Prediction Loss. To encourage each relation slot to be task-informative, *i.e.*, to carry evidence that supports answer prediction beyond merely signaling its identity, we add an auxiliary value objective. For each training example b, we take the activation of the gold relation slot $r_b = \text{rule_idx}_b$ and predict the first token of the answer via a relation-specific head:

$$\mathcal{L}_{\text{value}} = \frac{1}{B} \sum_{b=1}^{B} \text{CrossEntropy}\Big(\text{V}_{r_b}\big(z_{b,n_{\text{free}}+r_b}\big), t_b\Big), \tag{8}$$

where t_b is the first token of the ground-truth answer and V_r is a two-layer MLP mapping the scalar slot activation to vocabulary logits. We backpropagate $\mathcal{L}_{\text{value}}$ jointly with the reconstruction and binding losses so that concept slots remain predictive of the target output, while the large free-feature bank captures residual variation needed for high-fidelity reconstruction. The value heads serve as an auxiliary training signal (and diagnostic readout) and are not used in inference-time interventions.

Independence validation (not trained). Although we do not optimize an explicit free-slot independence term, we validate redundancy via an *independence score* computed as the squared off-diagonal covariance among free features:

$$\mathcal{L}_{\text{indep}} = \sum_{i \neq j} \left(\frac{1}{B} \sum_{b=1}^{B} (z_{b,i} - \bar{z}_i)(z_{b,j} - \bar{z}_j) \right)^2, \tag{9}$$

where $\bar{z}_i = \frac{1}{B} \sum_{b=1}^{B} z_{b,i}$. Without Stage 1, this score explodes (4.76→157.80, +3213.7%), indicating highly correlated and redundant free features.

C.4 Loss Weight Selection

Selection protocol. We did not perform an extensive hyperparameter search. Unless explicitly noted, we use **standard/default SAE choices** and keep them fixed across all runs. We only ran minimal sanity checks to avoid degenerate behavior (*e.g.*, dead features or collapse), and we did not tune weights to maximize reported metrics.

The loss components are weighted to balance competing objectives:

- $\lambda_{\text{recon}} = 1.0$ — High priority is given to faithful reconstruction to maintain model performance. This ensures the SAE does not distort the information flow through the network.
- $\lambda_{\text{sparse}} = 1 \times 10^{-3}$ — Gentle L1 penalty on free slots. This small weight prevents over-suppression of activations while still encouraging selective feature usage. Stronger sparsity penalties ($\lambda > 10^{-2}$) caused excessive dead neurons and degraded reconstruction quality in preliminary experiments.
- $\lambda_{\text{align}} = 1.0$ — Strong supervision signal to ensure reliable slot-relation binding. This weight is balanced with reconstruction to achieve $> 95\%$ binding accuracy on in-distribution templates in Fig. 4.
- $\lambda_{\text{ortho}} = 1 \times 10^{-2}$ — Moderate orthogonality constraint between relation and free slots. This maintains separation between supervised and unsupervised features, preventing information leakage that could compromise the interpretability of relation slots.
- $\lambda_{\text{value}} = 0.5$ — Balanced weight for answer prediction. This auxiliary task provides a training signal to ensure relation slots encode semantically meaningful information, but is weighted lower than alignment to avoid dominating the optimization.

C.5 Training Hyperparameters

Selection protocol. We use standard/default optimizer settings (AdamW) and do not tune hyperparameters for peak performance; we only ran minimal sanity checks for training stability.

Optimizer configuration. We use AdamW with the following settings (Loshchilov & Hutter, 2019):

- **Learning rate:** 1×10^{-3} (constant, no warmup or decay)
- **Weight decay:** 0.0 (L2 regularization disabled to avoid interfering with explicit sparsity constraints)
- **Betas:** $(0.9, 0.999)$ (default momentum coefficients)
- **Epsilon:** 1×10^{-8} (numerical stability constant)
- **Batch size:** 64 samples per update
- **Gradient clipping:** None (training was stable without clipping)

Training Schedule. The constant learning rate without decay was chosen because the two-stage training protocol naturally provides curriculum learning: Stage 1 establishes a good initialization for the encoder-decoder using traditional SAE framework Shu et al. (2025), after which Stage 2 refines the latent structure, as defined as SAE post-training. Preliminary experiments with cosine annealing showed no improvement over constant learning rate for this setting.

D Evaluation Metrics

This section provides detailed mathematical definitions for all evaluation metrics reported in Section 5 of the main paper.

D.1 Binding Accuracy Metrics

We evaluate the quality of semantic binding using multiple complementary metrics reported in Table 2 and Figure 3 of the main paper:

Slot Binding Accuracy. The fraction of questions that activate the correct relation slot, defined as:

$$\text{Acc}_{\text{binding}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\left[\underset{j}{\text{argmax}}\ z_{\text{rel},j}^{(i)} = r_i\right] \tag{10}$$

where $z_{\text{rel}}^{(i)}$ is the relation slot activation vector for question i and r_i is the ground-truth relation. This metric measures one-to-one mapping quality and is the primary metric reported in Table 2.

Top-k Accuracy. A relaxed metric checking whether the true relation slot appears in the top-k predictions:

$$\text{Acc}_{\text{top-}k} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}[r_i \in \text{TopK}(z_{\text{rel}}^{(i)})] \tag{11}$$

This metric is useful for understanding near-miss cases where the correct slot has high but not maximal activation.

Margin. The logit difference between the top-1 and top-2 slot predictions, measuring binding confidence:

$$\text{Margin} = \frac{1}{N} \sum_{i=1}^{N} (z_{\text{rel},j_1}^{(i)} - z_{\text{rel},j_2}^{(i)}) \tag{12}$$

where j_1 and j_2 are the indices of the highest and second-highest activations. Higher margins indicate more confident and unambiguous binding. We report average margins in Section 5.3 when analyzing binding robustness across layers.

Answer Accuracy. Exact-match accuracy for generated answers:

$$\mathrm{Acc}_{\mathrm{answer}} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}[\hat{a}_i = a_i] \tag{13}$$

where the normalization function handles multiple date formats (*e.g.*, *"Day, Month, Year"*; *"Month Day, Year"*; *"YYYY-MM-DD"*) to avoid penalizing formatting differences.

Swap Intervention Accuracy (Causal Control). Measures the full language model's answer generation after latent manipulation in swap experiments (Section 5.4). After modifying relation slot activations ($z_i^{\mathrm{orig}} \leftarrow 0$, $z_j^{\mathrm{target}} \leftarrow \alpha$), we decode to obtain $\hat{h} = W_{\mathrm{dec}} \cdot z'$ and feed this modified activation through the remaining transformer layers to generate text using the LLM's standard autoregressive generation. The value heads are *not used* in these experiments. This validates causal control over model behavior. Optimal swap success: 85% at $\alpha \approx 2$ (Layer 6).

The swap intervention accuracy validates that these representations causally influence the full model's behavior.

Diagonal Accuracy. Quantifies the one-to-one mapping quality between ground-truth relations and predicted slots using the confusion matrix:

$$\mathrm{Diag} = \frac{1}{R}\sum_{i=1}^{R} C_{ii} \tag{14}$$

where $R = 6$ is the number of relations and C is the normalized confusion matrix with $C_{ij} = \frac{\#\{r=i,\hat{r}=j\}}{\#\{r=i\}}$. Perfect binding yields Diag = 1.0, while random assignment gives Diag $\approx$ 0.167. Confusion matrices are visualized in Figure 4 for layer-wise analysis.

D.2 Reconstruction Quality

We measure the fidelity of the autoencoder's reconstruction using mean squared error:

$$\mathrm{MSE}_{\mathrm{recon}} = \frac{1}{N}\sum_{i=1}^{N} \|h^{(i)} - \hat{h}^{(i)}\|^2 \tag{15}$$

where $h^{(i)}$ is the original 768-dimensional activation vector from GPT-2's residual stream and $\hat{h}^{(i)} = W_{\mathrm{dec}} \cdot z^{(i)}$ is the reconstructed activation after encoding and decoding through the SAE. The reconstruction target is the raw activation, not normalized or preprocessed. Layer 6 achieves MSE $\approx 7.42 \times 10^{-2}$, representing a trade-off between reconstruction fidelity and semantic structure—early layers achieve lower MSE (*e.g.*, Layer 0: 6.53×10^{-5}) but lack meaningful concept binding. The relatively higher MSE in middle layers reflects the cost of enforcing interpretable slot structure while maintaining sufficient information for downstream task performance.

D.3 Swap Controllability

To test whether ontology-aligned slots serve as usable *control knobs*, we measure the success rate of answer swaps under inference-time steering along decoded SAE directions (protocol in §4.4; implementation details in §B.4).

Intervention. Let $h_{\ell,t} \in \mathbb{R}^d$ denote the residual stream at layer ℓ and token position t, and let $W_{\mathrm{dec}} \in \mathbb{R}^{d\times K}$ be the SAE decoder. For a target concept slot $j \in \{1,\ldots,K\}$, define the decoded direction $v_j := W_{\mathrm{dec}} e_j$. Given a question whose gold relation is $r^\star$ and a chosen target slot $j \neq \pi(r^\star)$, we steer the residual stream by

$$h'_{\ell,t} \;=\; h_{\ell,t} + \alpha v_j \;=\; h_{\ell,t} + W_{\mathrm{dec}}(\alpha e_j), \tag{16}$$

then continue the forward pass through the remaining transformer layers and decode the answer. We sweep amplification strengths

$$\alpha \in \{0.1, 0.5, 1, 2, 5, 10, 20, 50, 100, 200, 500, 1000\}. \tag{17}$$

Swap success. For swap trial m, let p_m be the subject entity and let r_m^{tgt} be the target relation corresponding to the amplified slot j. Let $\hat{a}_m^{\text{swap}}$ be the generated answer under the intervention. We count the swap as successful if the model outputs the gold value for the *target* relation, $g(p_m, r_m^{\text{tgt}})$:

$$\text{Swap}_\alpha = \frac{1}{M} \sum_{m=1}^{M} \mathbb{1}\big[\hat{a}_m^{\text{swap}} = g(p_m, r_m^{\text{tgt}})\big]. \tag{18}$$

Full metric definitions are provided in §D.

As shown in Figure 6, controllability is maximized at moderate amplification ($\alpha \approx 2$): at Layer 6, swap success rises from 0.54 at $\alpha = 1$ to 0.85 at $\alpha = 2$, but drops to 0.23 at $\alpha = 10$, indicating that over-amplification destabilizes the intervention.

E Layer-wise SAE Feature Comparison

This section presents a comprehensive comparison of top-50 activated features across all 12 transformer layers (Layer 0–11) of GPT-2. Each visualization compares two conditions: (1) **with SAE post-training** (supervised sparse autoencoder applied), and (2) **without SAE post-training** (baseline activations). This comparison reveals how supervised alignment shapes feature representations and demonstrates the emergence of clean semantic binding in middle layers, while also highlighting artifacts that appear in deeper layers without SAE regularization.

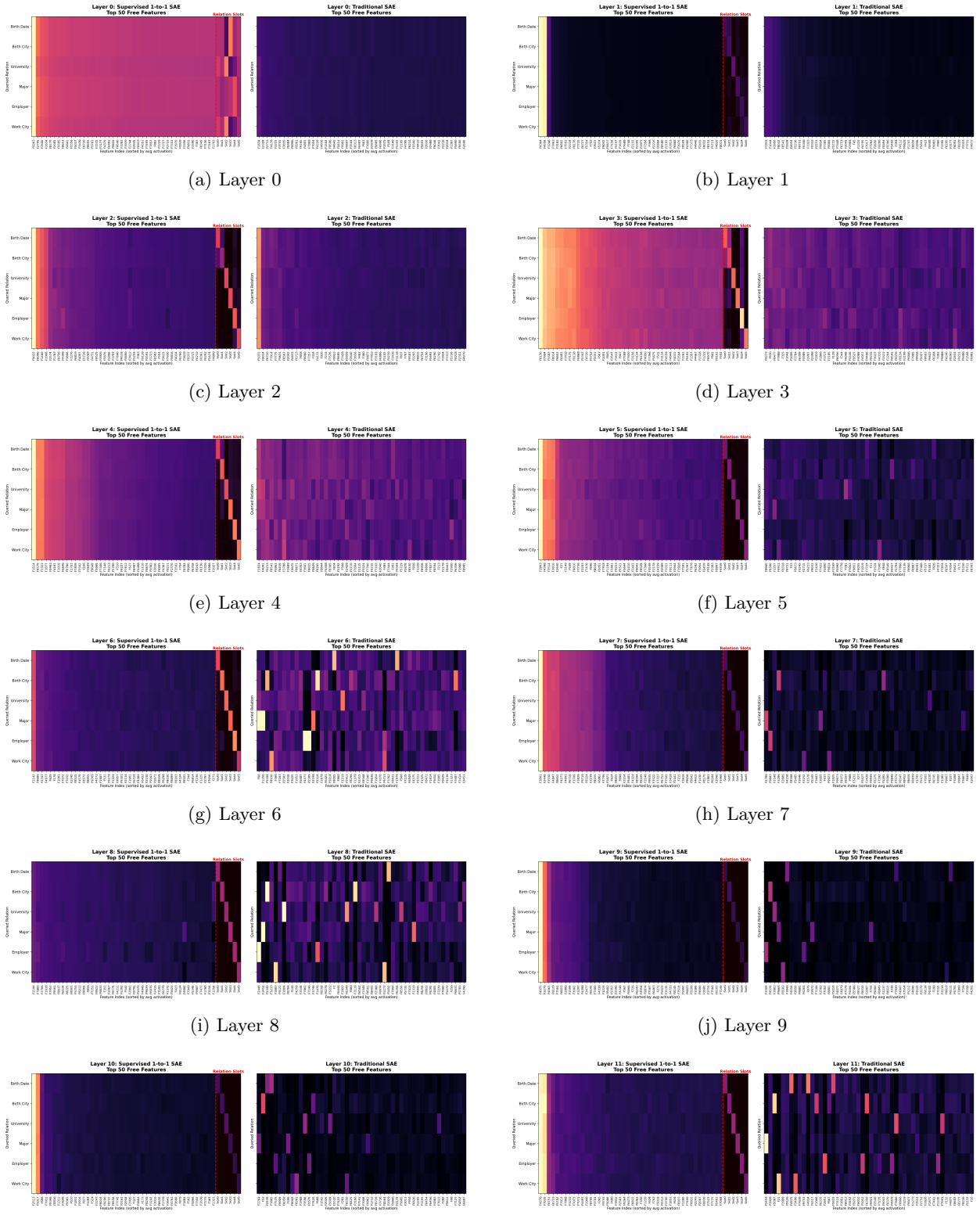
E.1 Analysis

These layer-wise visualizations reveal several critical insights about the role of supervised SAE training across network depth:

Early Layers (0–4): Limited Semantic Structure. In shallow layers, both conditions (with and without SAE) show relatively diffuse activation patterns with weak relation-specific structure. This reflects that early transformer layers primarily process local token-level features and have not yet formed abstract semantic representations suitable for clean concept binding. The SAE provides marginal improvements but cannot overcome the fundamental limitation that these layers lack the representational capacity for high-level semantic concepts.

Middle Layers (5–8): Emergence of Clean Binding. The most dramatic differences appear in middle layers, particularly Layer 6. With SAE post-training, we observe sharp, diagonal activation patterns indicating successful one-to-one binding between ontological relations and designated slots. Without SAE supervision, these same layers show more scattered, overlapping activations that fail to achieve clean separation between semantic concepts. This demonstrates that while middle layers contain the raw representational power for concept binding, explicit supervision through the SAE's multi-objective loss is essential to crystallize these latent capabilities into interpretable, controllable structure.

Deep Layers (9–11): Artifacts Without SAE. A particularly interesting phenomenon emerges in deeper layers without SAE post-training. Starting around Layer 6 and becoming more pronounced in Layers 9–11, the baseline (no SAE) condition exhibits strange, irregular activation patterns—potentially including sparse, extreme activations, feature collapse, or unexpected clustering. These artifacts likely reflect the model's aggressive compression of task-relevant information in preparation for final output generation. The supervised SAE mitigates these irregularities by enforcing reconstruction fidelity, sparsity constraints, and orthogonality between relation and free slots, resulting in more stable and interpretable features even in deep layers.



(a) Layer 0 (b) Layer 1

(c) Layer 2 (d) Layer 3

(e) Layer 4 (f) Layer 5

(g) Layer 6 (h) Layer 7

(i) Layer 8 (j) Layer 9

(k) Layer 10 (l) Layer 11

Figure 12: Layer-wise top-50 feature activations in GPT-2 with and without SAE post-training. Early layers (0–3) remain diffuse and entangled. Middle layers (4–8) show maximum SAE benefit, with Layer 6 achieving optimal semantic binding and diagonal accuracy. Deep layers (9–11) without SAEs exhibit irregular patterns and over-compression, whereas SAE-trained models maintain structured representations.

Orig → Swap	Question	Target	Generated
COMPANY_CITY → UNIVERSITY	What is Grace Wendy Rivera's work city?	Florida International University	Florida International University
COMPANY_CITY → MAJOR	What is Grace Wendy Rivera's work city?	Electrical Engineering	Electrical Engineering
COMPANY_CITY → EMPLOYER	What is Grace Wendy Rivera's work city?	Blackstone	Blackstone
UNIVERSITY → BIRTH_DATE	Where did Thomas Heath Stafford go to college?	2, March, 1981	2, March, 1981
UNIVERSITY → MAJOR	Where did Thomas Heath Stafford go to college?	Dance	Dance
BIRTH_DATE → WORK_CITY	When was Megan Kian Valencia born?	Framingham, MA	Framingham, MA
BIRTH_CITY → BIRTH_DATE	Where was Angela Maddox Gates born?	27, November, 1950	27, November, 1950
MAJOR → UNIVERSITY	What was Jennifer Donovan Pruitt's major?	University of Wisconsin–Madison	University of Wisconsin–Madison

Table 7: Correct swap examples from the evaluation set (Layer 6; α=2).

Optimal Layer for Intervention. These visualizations provide empirical justification for choosing Layer 6 as the primary layer for semantic intervention experiments (as reported in Section 5 of the main paper). Layer 6 achieves: (1) mature semantic representations that support clean binding, (2) strong separation between concepts under SAE training, (3) minimal artifacts compared to deeper layers, and (4) acceptable reconstruction error that preserves model functionality.

The Role of Supervised Training. Across all layers, SAE post-training consistently produces more structured, interpretable activation patterns. The supervised losses—particularly alignment loss (enforcing relation-slot correspondence) and orthogonality loss (separating supervised and unsupervised features)—act as powerful inductive biases that shape the latent space into a form amenable to human interpretation and causal intervention. Without this supervision, even layers with rich semantic content fail to expose that structure in an accessible format.

F Additional Qualitative Swap Examples

We provide qualitative swap examples to complement the quantitative controllability results (§D.3, §5.3). Unless otherwise noted, all examples use Layer 6 and intervention strength α=2. In each example, we start from a question whose gold relation is the *Orig* type and intervene by amplifying the decoded direction of a different concept slot (*Swap*), aiming to make the model answer with the corresponding target attribute.

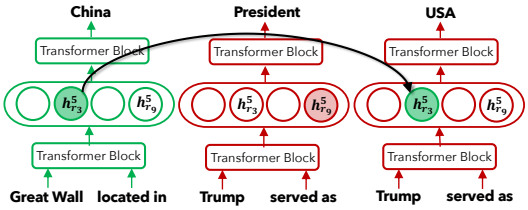


Figure 13: Illustrative schematic of our swap intervention (Orig relation → Swap relation).

F.1 Swapping Schematic

Fig. 13 illustrates the swap intervention: we steer the layer-ℓ residual representation along the decoded direction of a target (Swap) concept slot to flip the predicted answer type from the original relation.

F.2 Correct Swap Examples

Table 7 reports successful swaps. **Orig→Swap** denotes the original relation queried and the target relation we force via intervention. **Target** is the gold value $g(p,\text{Swap})$ for the same person p, and **Generated** is the model output after the swap (counted as correct when it matches the Target).

Two-hop example (step-wise supervision).

Query	Who is the `friend` of the `report` of `Avery`?
Canonical	$r_1 =$ `reports_to`, $r_2 =$ `friend_of`
Target	`Dominic Gerald` (*i.e.*, $e_2 =$ `Dominic`, $e_3 =$ `Gerald`)

Hop 1 evidence ($e_1 \xrightarrow{r_1} e_2$)
`Dominic reports_to Avery`

- "Dominic needs to send regular updates to Avery."
- "Dominic works under the supervision of Avery."
- "Avery is Dominic's manager at work."
- "Dominic answers directly to Avery."
- "Avery oversees Dominic's work tasks."
- "Dominic's reports go straight to Avery."

Hop 2 evidence ($e_2 \xrightarrow{r_2} e_3$)
`Gerald friend_of Dominic`

- "Gerald and Dominic are best pals."
- "Everyone knows that Gerald is a close friend of Dominic."
- "Gerald often hangs out with Dominic."
- "Dominic and Gerald are good friends."
- "Gerald is friendly with Dominic."
- "Gerald considers Dominic a trusted friend."

Aligned-slot supervision: when predicting e_2, the slot indicates r_1; when predicting e_3, it indicates r_2.

Figure 14: **Two-hop dataset example.** Input query, step-wise targets, and hop-specific profile evidence.

G Two-hop Reasoning Data Format

Dataset and setup. We further evaluate AlignSAE on a *two-hop* reasoning task adapted from prior work (Du et al., 2025). Each instance specifies a starting entity e_1 and an ordered pair of relations (r_1,r_2), and requires predicting a two-step chain (e_2,e_3) that forms the compositional path $e_1 \xrightarrow{r_1} e_2 \xrightarrow{r_2} e_3$, *i.e.*, $e_2 = r_1(e_1)$ and $e_3 = r_2(e_2)$. We instantiate the two-hop ontology with 20 relation types: `accuses`, `admires`, `blames`, `boss_of`, `classmate_of`, `competes_with`, `cousin_of`, `endorsed_by`, `follows`, `forgives`, `friend_of`, `has_crush_on`, `mentor_of`, `neighbor_of`, `owes_debt_to`, `protects`, `reports_to`, `subscribes_to`, `warns`, `works_with`.

Queries follow compositional templates such as "Who is the r_2 of the r_1 of e_1?", and are accompanied by short profile sentences that verbalize the hop-1 and hop-2 facts needed to answer the question. Concretely, each instance includes (i) hop 1 evidence describing the triple (e_1, r_1, e_2) and (ii) hop 2 evidence describing (e_2, r_2, e_3), each realized as multiple natural-language paraphrases (see Fig. 14).

We impose *step-wise* supervision (illustrated in Fig. 2): the model is trained to output the intermediate entity followed by the final entity (*i.e.*, "e_2 e_3"), so that slot alignment can be evaluated at each hop. Concretely, when generating e_2 the aligned concept slot should indicate r_1, and when generating e_3 it should indicate r_2. We generate 8,000 question–answer pairs in total and use a 4,000/4,000 train/validation split (no label noise). Based on the layer sweep in §D.3, we use layer 6 for all two-hop experiments.

Example. Fig. 14 shows one dataset instance. The question asks for the R_2-relation of the R_1-relation of the starting entity (*Avery*). The supporting profile sentences for hop 1 state that *Dominic reports_to Avery*, and the hop 2 profile sentences state that *Gerald* is a *friend_of Dominic*. The step-wise target output is therefore *"Dominic Gerald"*.

Training hyperparameters. We train the base GPT-2 (124M; 12 layers, 768-d) with AdamW and a warmup+cosine learning-rate schedule. We did **not** conduct an extensive hyperparameter search; instead, we adopt a widely used configuration for GPT-style fine-tuning (AdamW with weight decay, linear warmup, cosine decay, and moderate batch sizes) and keep it fixed across runs, only sanity-checking for stable convergence (no divergence). Concretely, we use an effective batch size of 96, maximum sequence length 512, and train for 100,000 update steps with learning rate warmed up for 2,000 steps to a peak of 5×10^{-4} and then cosine-decayed to a minimum of 1×10^{-5}. We set weight decay to 0.1 and Adam ϵ to 1×10^{-8}. The training performance for the two-hop task is illustrated in Figure 15.

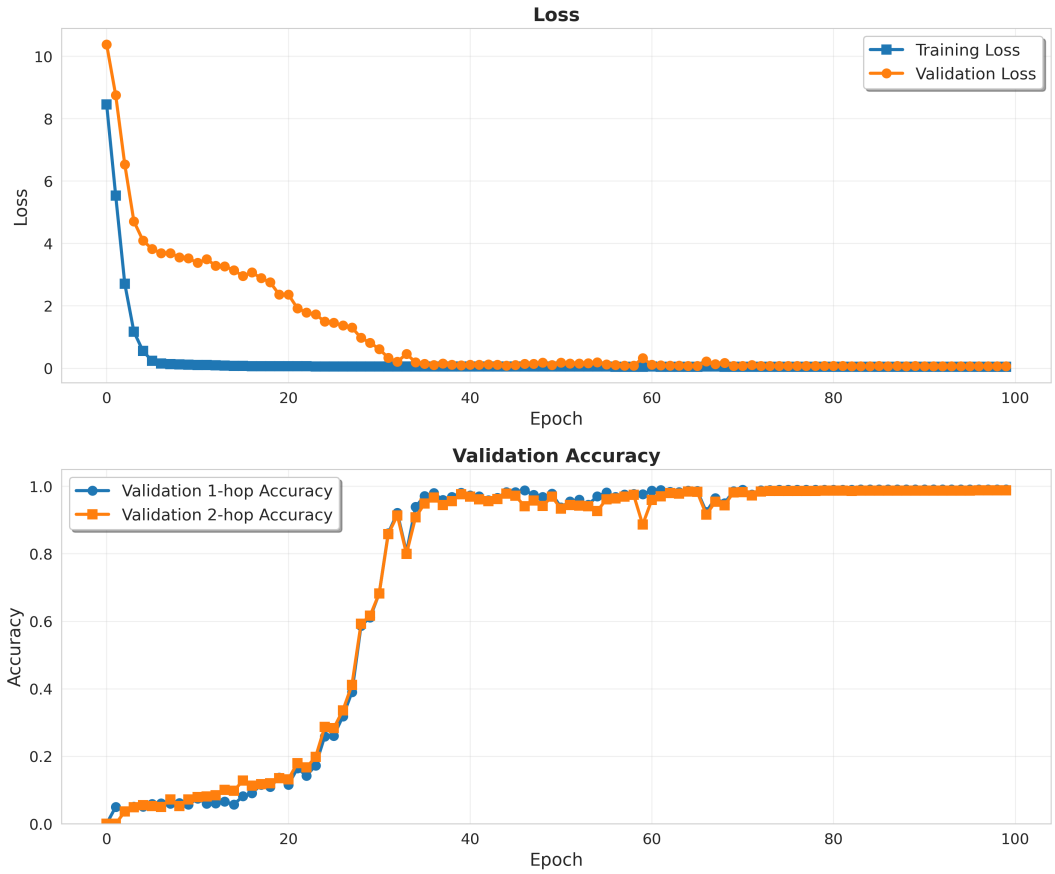


Figure 15: Training dynamics on the 2-hop task. Top: training vs. validation loss. Bottom: validation accuracy for the 1st hop (Token 1, e_2) and the 2nd hop (Token 2, e_3) under step-wise supervision.