AlignSAE: Concept-Aligned Sparse Autoencoders

Minglai Yang[1] **Xinyu Guo**[1] **Mihai Surdeanu**[1] **Liangming Pan**[2*]
[1]University of Arizona [2]MOE Key Lab of Computational Linguistics, Peking University
{mingly, xinyuguo, msurdeanu}@arizona.edu
liangmingpan@pku.edu.cn

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 introduce *AlignSAE*[1], a method that aligns SAE features with a defined 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 relations 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.

1 Introduction

While Large Language Models (LLMs) have rapidly advanced in capability, their internal mechanisms remain largely opaque. Mechanistic interpretability (Bereska and Gavves, 2024; Saphra and Wiegreffe, 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 known as 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, individual neurons become difficult to interpret, as their activations represent entangled mixtures of distinct 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, 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 birth cities. However, because standard SAEs are trained in an unsupervised fashion, it has no explicit incentive to align its 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 and Gavves, 2024; Shu et al., 2025). 2) *Features remain entangled*: concepts are often fragmented across multiple features, and conversely, a single feature may respond to multiple unrelated concepts, as shown in Figure 1 (left). These limitations undermine downstream applications that require reliable feature-level control, such

*: Corresponding Author
[1]Work is still in progress.

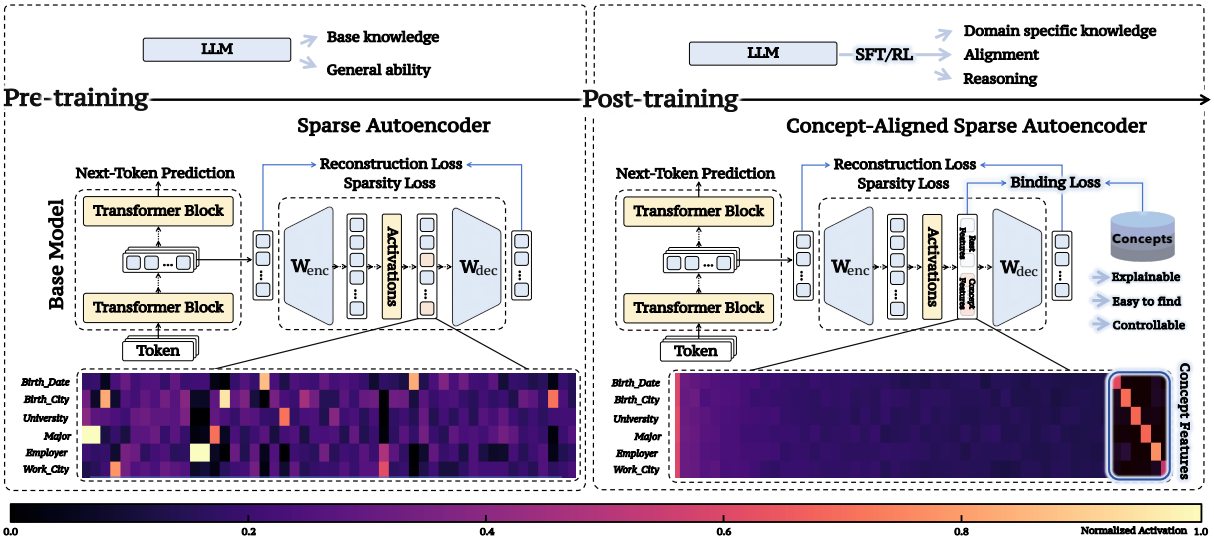


Figure 1: An overview of our approach. Left: An unsupervised SAE trained post hoc on frozen LLM activations optimizes only reconstruction and sparsity, so each concept tends to be spread across multiple features, making interventions unreliable. Right: Our Concept-Aligned SAE adds a supervised binding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

as safety steering (Bereska and 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; Paulo and Belrose, 2025).

To mitigate these issues, we take inspiration from the training pipeline of LLMs. As illustrated in Figure 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 introduces guidance on top of the unsupervised SAE. The goal explicitly is 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 a large SAE to one layer of a frozen base LM and train it in two phases: first unsupervised, then supervised. After the SAE has learned a general reconstruction-oriented code (the pre-training phase), we "fine-tune" the SAE with concept supervision. We designate K special latent feature slots in the autoencoder, each slot corresponding to a specific target concept from a given knowledge ontology, while the remaining dimensions form a free feature bank to preserve overall 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 Figure 1 (right).

Empirically, our Concept-Aligned SAE yields a feature representation that is significantly more interpretable, disentangled, and controllable than a standard SAE baseline. Each target concept in our experiments cleanly maps to its own single feature (Figure 3 & Figure 4). The concept-aligned features exhibit monosemantic behavior, activating precisely for their concept across varied phrasings and contexts, which yields improved generalization to unseen prompt templates. Moreover, we find that we can reliably steer the model's outputs by intervening on these features: toggling a concept's slot activation causes the model to add, emphasize,

or suppress information about that concept in its response, with minimal side effects on unrelated content (Figure 5 & Table 3).

2 Related Work

We overview two main directions that influenced this work, *Sparse Autoencoder Steering* and *Concept Binding*, which focus on aspects of interpretable control and concept alignment, respectively. Our work takes inspiration from these directions, introducing a lightweight, concept-aligned, and interpretable SAE framework.

Sparse Autoencoder Steering. Sparse Autoencoders (SAEs) provide an interpretable, lightweight 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). A growing body of work improves feature quality through training objectives and architectural choices (*e.g.*, JumpReLU, Top-k sparsification, and better dictionary structure) (Rajamanoharan et al., 2024; Bricken et al., 2023; Cunningham et al., 2023; Shu et al., 2025; Sharkey et al., 2025). However, because SAE features are learned purely unsupervised, 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). To address this, we introduce an SAE post-training stage with concept supervision that *learns the concept-to-feature mapping during post-training*: we reserve dedicated concept slots, bind each ontology concept to a fixed slot, and thereby make interventions targetable and reproducible without post-hoc feature hunting.

Concept Binding. Posterior Regularization (PR) (Ganchev et al., 2010) and Logic Rule Encoding (LRE) (Hu et al., 2016) are two traditional frameworks widely adopted to bind human-defined concepts to neural models by imposing soft constraints on posterior distribution, or integrating first-order 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 sentence-level 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 logical rule loss, thereby binding logic-level concepts to model predictions via parameter updates. Although LRE addresses the soft-constraint issue of PR and yields stronger empirical performance, it still remains a black-box mechanism that cannot be used to interpret or control specific internal representations of the model. Concept Bottleneck Models (CBM) (Koh et al., 2020) were proposed to transform the outputs of intermediate layers of base models into human-understandable concepts by adding a concept-mapping loss to the training objectives. 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 Overview

Our primary goal is to transform the implicit, distributed knowledge of a Large Language Model (LLM) into an explicit, verifiable, and controllable interface. We achieve this by training a *Concept-Aligned Sparse Autoencoder (SAE)* based on the activations of a frozen LLM.

Terminology In this work, we instantiate *concepts* as relation types drawn from a domain ontology. By an *ontology*, we mean a finite inventory of relation types that capture semantic links between entities (*e.g.*, BORN_IN, FRIEND_OF, WORK_CITY). A *relation* is one such type, and each relation mention in text is represented as a triple (e_1, r, e_2), where r is the relation type and e_1, e_2 are the participating entity mentions—for example, $\big(\text{"Marie Curie"}, \text{BORN\_IN}, \text{"Warsaw"}\big)$. We treat these relation types as atomic concepts for convenience; our framework is agnostic to this choice and could equally bind other forms of domain knowledge (*e.g.*, attributes, events, or scientific categories) to supervised SAE slots.

Concept Binding in Activation Space Let the model's knowledge be represented by a set of triples $\mathcal{K} = \{(x, r, y)\} \subset \mathcal{E} \times \mathcal{R} \times \mathcal{V}$, where $\mathcal{E}$ is a set of entities, $\mathcal{R}$ is a finite set of relations, and $\mathcal{V}$ is a set of values. Let $\mathcal{I} : \mathcal{E} \times \mathcal{R} \to \mathcal{X}$ be a prompt generation function mapping a subject-relation pair to a natural language query $q \in \mathcal{X}$. We consider a frozen language model M and denote the activation vector at layer ℓ for input q as $h = M_\ell(q) \in \mathbb{R}^d$. We posit that h encodes the relation r in a recoverable subspace.

Our objective is to learn a sparse decomposition $z = E(h) \in \mathbb{R}^K$ via a Sparse Autoencoder (SAE). We explicitly partition the latent features into two sets: *concept features* z_{concept} and *monosemantic features* z_{rest}. The concept features are supervised to align one-to-one with our ontology $\mathcal{R}$, while the monosemantic features remain unsupervised to capture other distributional statistics. Formally, we seek an injective mapping $\pi : \mathcal{R} \to \{1, \ldots, K\}$ identifying the indices of concept features. For any query $q = \mathcal{I}(x, r)$, we enforce that the specific concept feature $z_{\pi(r)}$ is active (i.e., $z_{\pi(r)} > \tau$), while other concept features $z_{\pi(r')}$ for $r' \neq r$ are suppressed. This ensures that $z_{\pi(r)}$ serves as a verifiable indicator for relation r.

A Verifiable Interface Unlike standard probing, 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.

We apply this method to a single intermediate layer of the LLM, effectively treating it as a read and write head for the model's internal state. The following sections detail the architecture of this interface (§4) and its validation on a biography reasoning task from a knowledge graph (§5).

4 Implementation

We build on a frozen decoder-only transformer, instantiated as GPT–2 for concreteness. Given an input biography question x, the model yields token-level hidden states at each layer. We extract a pooled representation $h \in \mathbb{R}^d$ from a fixed intermediate layer ℓ (*e.g.*, mean over the question span). Freezing the language model preserves its general linguistic competence and places all supervision on a light-weight interface trained on top of these activations.

4.1 Concept-Aligned Sparse Autoencoder

We introduce a large supervised sparse autoencoder (SAE) that exposes an interpretable control surface aligned with the ontology $\mathcal{R}$ while delegating remaining variance to a large bank of unsupervised features. The encoder $E : \mathbb{R}^d \to \mathbb{R}^K$ maps h to a sparse code $z = \text{ReLU}(W_e h + b_e)$. We partition z as $z = [z_{\text{concept}}; z_{\text{mono}}]$, where $z_{\text{concept}} \in \mathbb{R}^{|\mathcal{R}|}$ are supervised concept slots and $z_{\text{mono}} \in \mathbb{R}^{K-|\mathcal{R}|}$ are unsupervised monosemantic features. The decoder $D : \mathbb{R}^K \to \mathbb{R}^d$ reconstructs $\hat{h} = W_d z + b_d$. We use a large K (*e.g.*, $K \approx 100k$) to give the model ample capacity without burdening the $|\mathcal{R}|$ interpretable slots. A light value head $V : \mathbb{R}^{|\mathcal{R}|} \to \mathbb{R}^C$ consumes only z_{concept} to score candidate answers. This design aims to keep concept slots clean, addressable, and directly useful for prediction, while the monosemantic bank preserves reconstruction quality and absorbs nuisance factors.

4.2 Objectives

We train the encoder, decoder, and value head jointly. Our objective extends the standard SAE loss with (i) a binding term that assigns each ontology relation to a dedicated concept slot, (ii) a decorrelation penalty that reduces leakage of relation signal into non-concept features, and (iii) a value loss that forces the same concept slots to support answer prediction.

$$\mathcal{L}_{\text{SAE}} = \lambda_{\text{rec}} \|h - \hat{h}\|_2^2 + \lambda_{\text{sp}} \|z\|_1, \tag{1}$$

$$\mathcal{L}_{\text{bind}} = \text{CE}\big(\text{softmax}(z_{\text{concept}}), y_{\text{rel}}\big), \tag{2}$$

$$\mathcal{L}_{\perp} = \big\|\text{corr}(z_{\text{concept}}, z_{\text{rest}})\big\|_F^2, \tag{3}$$

$$\mathcal{L}_{\text{val}} = \text{CE}\big(\text{softmax}(V(z_{\text{concept}})), y_{\text{ans}}\big), \tag{4}$$

$$\mathcal{L} = \mathcal{L}_{\text{SAE}} + \lambda_{\text{bind}}\mathcal{L}_{\text{bind}} + \lambda_{\perp}\mathcal{L}_{\perp} + \lambda_{\text{val}}\mathcal{L}_{\text{val}}. \tag{5}$$

Here CE denotes cross-entropy and $\text{corr}(\cdot,\cdot)$ is a mini-batch correlation estimate. $\mathcal{L}_{\text{bind}}$ makes relation identity directly readable from the concept slots (i.e., selecting the intended slot recovers y_{rel}), while $\mathcal{L}_{\text{val}}$ forces these same slots to be sufficient for predicting the answer label via $V(\cdot)$, creating a consistent slot-level bottleneck that can be validated by intervention. The weak penalty $\mathcal{L}_{\perp}$ discourages correlation between z_{concept} and z_{rest}, reducing leakage of relation information into non-concept (distributional) features and stabilizing the

semantics of the designated slots. At the scale of K=100k features, we do not decorrelate within z_{rest} for efficiency; concept–rest decorrelation is sufficient in practice. Details are provided in Appendix C.3.

4.3 SAE Pre-training and Post-training

Directly optimizing the full objective from scratch can produce brittle or unstable slot formation, so we adopt a two-phase curriculum that parallels LLM pre-training and post-training. In the *pre-training* phase, the SAE is trained primarily on reconstruction and sparsity with only a weak binding signal, allowing the decoder to form a stable, high-capacity dictionary before any semantic commitments are imposed. In the subsequent *post-training* phase, we strengthen the binding and value losses and activate the orthogonality penalty, which systematically reshapes the latent space so that 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 cohered.

5 Experimental Setup

We validate our approach on a biography question answering task over a fixed, small ontology. Complete implementation details, including dataset generation procedures, model training hyperparameters, SAE architecture specifications, and evaluation metric definitions, are provided in Appendices A–D.

Ontology and Dataset Let $\mathcal{R} = \{$BIRTH_DATE, BIRTH_CITY,UNIVERSITY,MAJOR,EMPLOYER, WORK_CITY$\}$ denote six atomic relations. 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). An input x is a natural-language question mentioning p and implicitly targeting one $r^\star \in \mathcal{R}$.

We generate 1,000 synthetic person profiles with 5 biography variants each, drawn from a vocabulary of 411 first names, 461 middle names, 1,002 last names, 341 universities, 101 academic majors, and 327 companies (see Appendix A for complete entity vocabulary details). The dataset is constructed to separate *semantic binding* from *template memorization*. Question surfaces are generated from paraphrase templates that vary syntax and lexical cues while preserving the underlying relation. We partition templates into disjoint sets to induce two regimes (see Appendix A.3):

- **Training:** Uses 2 templates mixed with various persons.
- **Test-Unseen-Template:** Holds out 2 templates never seen during training

This protocol ensures that high scores require relation-level generalization rather than reliance on surface form artifacts. The full question-answer template set is detailed in Appendix A.

Model Training We fine-tune GPT-2 on biography memorization and question-answering tasks for 80,000 steps with batch size 96 (see Appendix B for details). Hidden states are extracted from the residual stream at the final question token across all 12 layers.

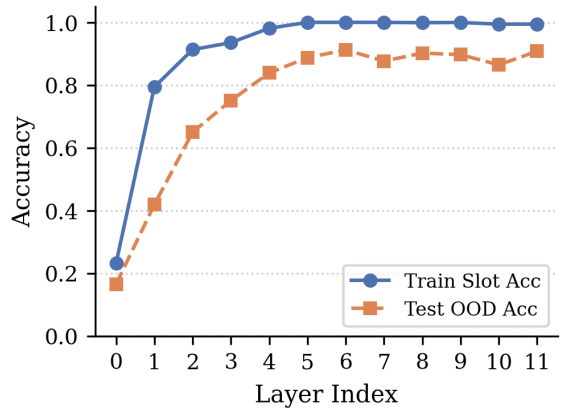
Our supervised SAE consists of 100,000 unsupervised free slots plus 6 supervised relation slots, trained in two stages: 50 epochs of reconstruction-only followed by 100 epochs with the full multi-objective loss combining reconstruction, sparsity, alignment, independence, orthogonality, and value prediction terms (Appendix C provides the complete loss function formulation and hyperparameter justification).

Metrics We evaluate relation binding by comparing the predicted concept feature $\hat{r}_i = \arg\max_j z^{(i)}_{\text{concept},j}$ with $r_i^\star$:

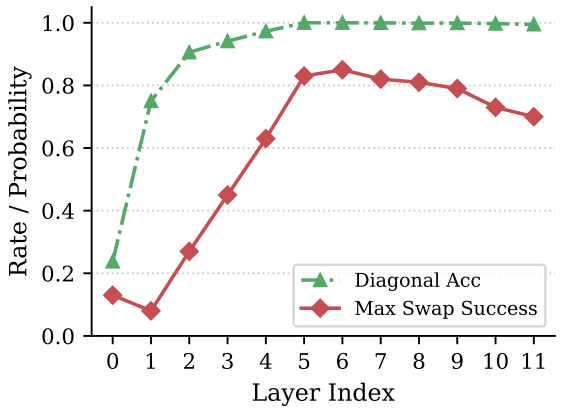
$$\text{Acc}_{\text{bind}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}[\hat{r}_i = r_i^\star] \,.$$

To assess one-to-one quality independent of slot permutations, we report *diagonal accuracy*, i.e., the fraction of mass on the diagonal of the relation–concept feature confusion matrix after a single global permutation is fixed from validation. We additionally use an *Unseen Template binding* score computed on the held-out paraphrase set to quantify phrasing-invariant concept capture.

For controllability, we evaluate a *swap test* in which a question targeting $r^\star$ is perturbed at inference time by injecting a decoded basis $D(e_j)$ for another relation $j \neq r^\star$ with strength $\alpha > 0$.



(a) Train slot accuracy vs. test unseen-template accuracy.



(b) Diagonal accuracy vs. swap success.

Figure 2: Comparison of binding generalization and causal intervention mechanisms.

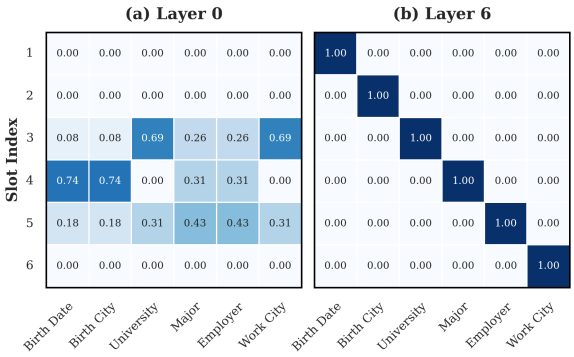


Figure 3: Relation–slot binding at a shallow layer (a) versus a mid layer (b) of GPT-2. At layer 0, supervision for each relation is dispersed across multiple slots, whereas at layer 6 the SAE learns a perfect one-to-one, diagonal binding, indicating that mid-layer representations are far more amenable to clean, controllable relation binding.

A swap is counted as successful if the generated answer switches to the gold value $g(p,j)$ under the perturbation. Formal definitions of all metrics including top-k accuracy, margin, answer accuracy, reconstruction quality, and swap controllability are provided in Appendix D.

6 Results

We evaluate the proposed interface across transformer layers, data regimes (Train, Test–Unseen-Template), and controllability settings. Metrics follow Section 5. All language-model parameters are frozen; only the SAE and value head are trained.

6.1 Layer-wise performance

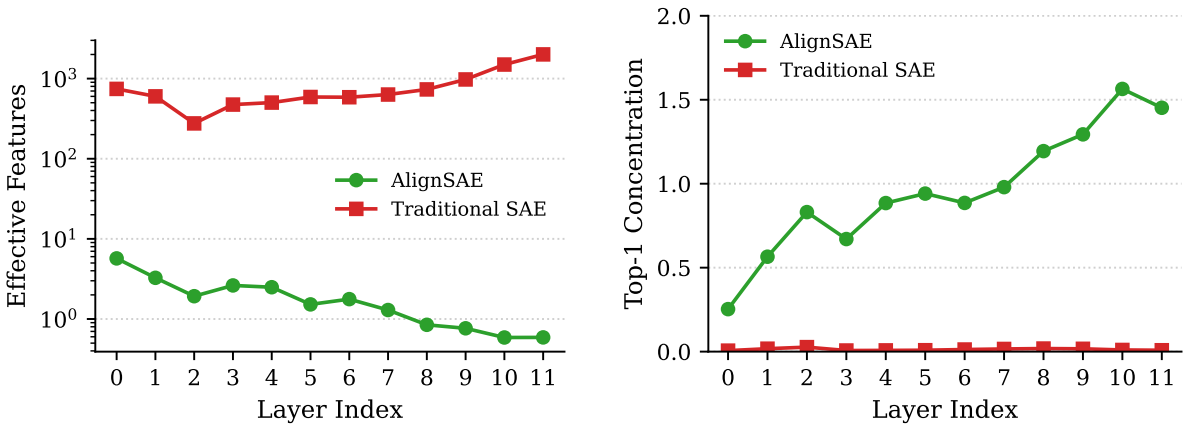
Figure 2a summarizes the layer sweep. Performance peaks in the middle of the stack: at layer 6 the model attains perfect one-to-one binding (*diagonal accuracy* $= 1.00$), strong swap controllability (*swap success* $= 0.85$ at $\alpha \approx 2$), and strong generalization to paraphrases (*Test-Unseen-Template slot acc* $= 0.912$), while maintaining faithful reconstructions. Early layers underperform (*e.g.*, layer 0 diagonal accuracy $= 0.238$, swap success $= 0.08$), consistent with representations that are too local to support concept-aligned slots. Deeper layers exhibit increased reconstruction error, suggesting heavier task compression that makes clean slot interfaces harder to preserve.

Metric	Layer 0	Layer 6	Δ (L6 – L0)
Diagonal Accuracy	0.238	**1.000**	↑ 0.76
Swap Success	0.040	**0.850**	↑ 0.81
Train Slot Acc	0.232	**1.000**	↑ 0.77
Test Unseen Acc	0.165	**0.912**	↑ 0.75
Recon MSE	6.53×10^{-5}	7.42×10^{-2}	↑ ≈ 1.1k×

Table 1: Performance comparison between Layer 0 (early) and Layer 6 (middle). Higher is better for accuracy metrics; lower is better for reconstruction MSE.

6.2 Structure of binding: early vs. middle layers

The contrast between layer 0 and layer 6 illustrates how concept alignment emerges. 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. Figures 3 visualize this effect.



(a) Concept fragmentation (lower is better).

(b) Concept concentration (higher is better).

Figure 4: Layer-wise concept fragmentation and concentration for AlignSAE (post-training) and a traditional SAE (pre-training only).

Table 1 highlights the massive difference in performance between early and middle layers. Layer 0 shows poor binding accuracy with scattered, overlapping slots, while Layer 6 demonstrates perfect 1-to-1 slot activation. The reconstruction MSE at Layer 6 is significantly higher than Layer 0 (7.42×10^{-2} vs 6.53×10^{-5}), but this trade-off yields a +76.2% improvement in diagonal accuracy and a +81% improvement in swap success. This suggests that semantic concept binding emerges in middle transformer layers where rich contextual representations are developed, rather than in early layers that process raw token embeddings.

6.3 Layer-wise concept fragmentation and alignment (Pre-training vs. Post-training)

To quantify how cleanly each concept is represented at different depths, we compare a traditional SAE trained purely unsupervised on pre-training activations with AlignSAE, which adds concept-supervision as a post-training signal. Let $z_i \in \mathbb{R}^K$ 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 as

$$A_{c,k} = \mathbb{E}_{i:c(i)=c}\big[z_{i,k}\big], \tag{6}$$

and normalize over features to obtain a concept–feature distribution

$$B_{c,k} = \frac{A_{c,k}}{\sum_{k'} A_{c,k'} + \epsilon}. \tag{7}$$

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 $B_{c,\cdot}$:

$$\mathrm{EffFeat}(c) = \exp\Big(-\sum_k B_{c,k} \log B_{c,k}\Big), \tag{8}$$

where smaller values indicate that a concept is concentrated on fewer features (i.e., lower fragmentation).

Top-1 concentration (Top1Conc). We also track how much of a concept's mass is captured by its single most responsive feature:

$$\mathrm{Top1Conc}(c) = \max_k B_{c,k}, \tag{9}$$

where larger values indicate that one feature clearly dominates the representation of concept c.

Figure 4 shows layer-wise averages of these metrics across six concepts for AlignSAE and the traditional SAE. The traditional SAE (pre-training only) exhibits extreme fragmentation at every layer: EffFeat is on the order of hundreds to thousands of features per concept and Top1Conc remains near zero, indicating that no single feature clearly encodes any concept. In contrast, AlignSAE uses post-training supervision to dramatically reduce fragmentation (EffFeat typically ≈ 1) and achieve much higher concentration. From mid layers onward, each concept is represented by a small, compact set of features; in the deepest layers, AlignSAE approaches near one-to-one bindings, where a single feature dominates each concept. This layer-wise analysis confirms that adding a concept-level

Orig → Swap	Question	Target (swap)	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 2: Correct swap examples from our evaluation set (Layer 6; α=2 throughout).

post-training signal turns the SAE feature space from a diffuse, many-to-many mapping into a compact, interpretable interface where specific concepts are easy to find and control.

6.4 Inference-Time Interface

Because each concept slot decodes to a direction in representation space, the interface naturally exposes an interpretable control: given relation index j and strength $\alpha > 0$, we may form $h' = \alpha$, where e_j selects the j-th concept slot. This optional steering, used only for qualitative analyses, illustrates that the ontology-aligned slots are addressable and can modulate answer type without modifying the base language model. Further details about layer selection, pooling, and training schedules appear in the Appendix B.

6.5 Controllability

We probe whether ontology-aligned slots serve as usable control knobs by amplifying decoded directions during inference. Figure 5 shows swap success across layers and amplification strengths. Moderate amplification ($\alpha \approx 2$) reliably switches the answer type at layers 5-11. For example, at Layer 6:

- $\alpha = 1.0$: Success Rate = 0.54. The model begins to switch but is not fully consistent.
- $\alpha = 2.0$: Success Rate = 0.85. The model reliably switches to the target answer.
- $\alpha = 10.0$: Success Rate = 0.23. Over-amplification causes instability.

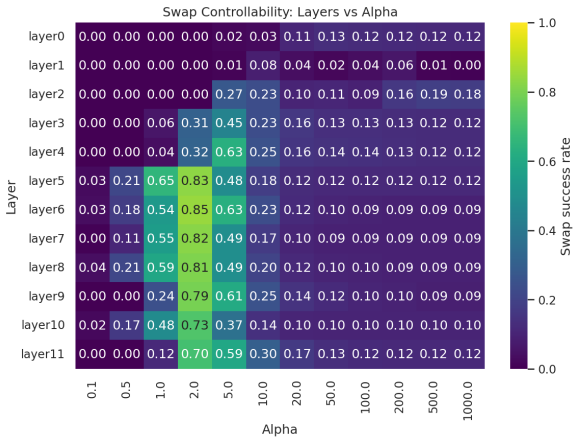


Figure 5: Swap controllability across layers (rows) and amplification α (columns). Lighter is better; mid layers sustain robust control around $\alpha \approx 2$.

These results indicate a specific operating range in which the interface is both effective and predictable.

6.6 Qualitative swaps

Qualitative examples corroborate the quantitative trends. For a birth-date question about *Reginald Deandre Barber* ("What is Reginald Deandre Barber's birth date?"), the model originally answers "24, March, 1964". By amplifying the UNIVERSITY slot with α=2, the generated answer switches to *Wesleyan University*, demonstrating that slots are not merely diagnostic but causal control handles. This confirms that the SAE has learned separate, controllable representations for different semantic concepts. Please see Table 2 for more examples.

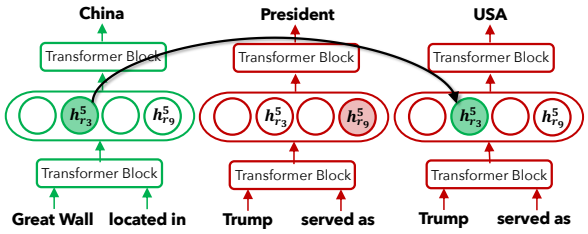


Figure 6: An illustrative example for swapping.

Swap	**Q:** Where did Jesse Kian Tate go to college? **Original:** UNIVERSITY → **Swap to:** MAJOR
Outputs	**Baseline:** Rochester Institute of Technology **Gold target:** Physical Therapy **Generated: Geography** *(type ✓ entity ✗)*

Table 4: A failure case: steering (α=10) flips the answer *type* correctly (to a major) but misses the exact entity.

6.7 Error analysis

Even when swap steering fails to hit the *exact* gold entity, it often preserves the *answer category* of the intended swapped relation (city/date/university/major/employer). Table 3 reports *Category Preservation*, computed *only over failed swaps*. Under moderate amplification (α=2), 74.7% 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 *type manifold* even as exact entity selection degrades.

Concretely, in the example (Table 4) we swap a UNIVERSITY question into the MAJOR slot at α=10. The model does not output the gold major (*Physical Therapy*), but it still outputs a plausible major (*Geography*), i.e., the steering succeeds at changing *what kind of attribute* is produced while failing at the finer-grained *which entity* decision.

Target swap	α=2 **(Error 15%)**			α=10 **(Errors 77%)**		
	Same	**Diff**	**Same %**	**Same**	**Diff**	**Same %**
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	**115**	**39**	**74.7**	**636**	**130**	**83.0**

Table 3: Category retention on *failed* swaps at Layer 6. "Same" means the generated answer falls in the correct semantic class for the swapped relation (even if the entity is wrong); "Diff" means it falls outside that class.

6.8 Takeaways

Middle layers furnish the cleanest substrate for ontology-aligned slots: binding is perfect, controllability is reliable under moderate $\alpha \approx 2$, and reconstruction remains acceptable. Early layers lack sufficient abstraction, and very deep layers trade off reconstruction for compressed task features, complicating clean interfaces. The results suggest that post-training concept alignment at mid layers is a viable path toward explainable and controllable world-knowledge access in frozen LMs.

7 Conclusion

We presented a practical route toward adding world models to LLMs by encoding ontological knowledge into a frozen model's mid-layer through a concept-aligned *SAE post-training* interface. Training on verifiable reasoning traces ties slot identity to ontology relations and makes answers predictable from those slots alone, yielding an interface that is simultaneously explainable and controllable. Across multiple small ontologies, we observed robust relation binding under paraphrase, reliable slot-level interventions that steer answers via decoded directions, and a consistent sweet spot in the middle layers where one-to-one alignment emerges most cleanly.

This approach shifts the focus from probing distributed representations to *operating* an addressable world-knowledge interface without changing base LM weights. In doing so, it narrows the gap between "no world model" and actionable structure: named concept slots act as stable variables that can gate tools or condition decisions in agents and domain-specific robots. While full dynamical world models remain a longer-term goal, our results indicate that modular, post-hoc concept alignment is a viable first step—turning shallow, distributed encodings into auditable, steerable handles for knowledge manipulation.

Future directions include scaling to richer, hierarchical ontologies, coupling slots to external memories and tools for closed-loop planning; enforcing cross-slot consistency and causal constraints; and extending beyond first-token targets to structured, multi-step answers. These avenues move from static alignment toward interactive, verifiable world modeling grounded in controllable interfaces.

Limitations

Our method is currently evaluated only on single-hop factual queries, where each question corresponds to a single concept feature. Extending the approach to multi-hop reasoning remains an open challenge. In particular, modeling interactions between multiple concept features across different layers of the LLM, such as composing relations or tracing multi-step inference chains, is still in progress. We leave the development of multi-hop concept binding and cross-layer reasoning circuits for future work.

Acknowledgement

We gratefully acknowledge support from the University of Arizona Undergraduate Research Travel Grant, which provided funding for Minglai Yang. We also thank the College of Information for additional student research funding and the AI Club at University of Arizona for their support.

References

Dana Arad, Aaron Mueller, and Yonatan Belinkov. 2025. Saes are good for steering – if you select the right features. *Preprint*, arXiv:2505.20063.

Reza Bayat, Ali Rahimi-Kalahroudi, Mohammad Pezeshki, Sarath Chandar, and Pascal Vincent. 2025. Steering large language model activations in sparse spaces. *Preprint*, arXiv:2503.00177.

Leonard Bereska and Stratis Gavves. 2024. Mechanistic interpretability for AI safety - a review. *Transactions on Machine Learning Research*. Survey Certification, Expert Certification.

Amrita Bhattacharjee, Shaona Ghosh, Traian Rebedea, and Christopher Parisien. 2024. Towards inference-time category-wise safety steering for large language models. *Preprint*, arXiv:2410.01174.

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

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. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*. https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Sviatoslav Chalnev, Matthew Siu, and Arthur Conmy. 2024. Improving steering vectors by targeting sparse autoencoder features. *Preprint*, arXiv:2411.02193.

David Chanin, Tomáš Dulka, and Adrià Garriga-Alonso. 2025. Feature hedging: Correlated features break narrow sparse autoencoders. *Preprint*, arXiv:2505.11756.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. 2023. Sparse autoencoders find highly interpretable features in language models. *Preprint*, arXiv:2309.08600.

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. 2022. Toy models of superposition. *Transformer Circuits Thread*. https://transformer-circuits.pub/2022/toy_model/index.html.

Eoin Farrell, Yeu-Tong Lau, and Arthur Conmy. 2024. Applying sparse autoencoders to unlearn knowledge in language models. *Preprint*, arXiv:2410.19278.

Javier Ferrando, Gabriele Sarti, Arianna Bisazza, and Marta R. Costa-jussà. 2024. A primer on the inner workings of transformer-based language models. *Preprint*, arXiv:2405.00208.

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

Kuzman Ganchev, João Graça, Jennifer Gillenwater, and Ben Taskar. 2010. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*, 11(67):2001–2049.

Shaona Ghosh, Amrita Bhattacharjee, Yftah Ziser, and Christopher Parisien. 2025. Safesteer: Interpretable safety steering with refusal-evasion in llms. *Preprint*, arXiv:2506.04250.

Phillip Guo, Aaquib Syed, Abhay Sheshadri, Aidan Ewart, and Gintare Karolina Dziugaite. 2024. Mechanistic unlearning: Robust knowledge unlearning and editing via mechanistic localization. *Preprint*, arXiv:2410.12949.

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. 2024. Llama scope: Extracting millions of features from llama-3.1-8b with sparse autoencoders. *Preprint*, arXiv:2410.20526.

Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. 2016. Harnessing deep neural networks with logic rules. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2410–2420, Berlin, Germany. Association for Computational Linguistics.

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. 2024. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*.

Yi Jing, Zijun Yao, Hongzhu Guo, Lingxu Ran, Xiaozhi Wang, Lei Hou, and Juanzi Li. 2025. Lingualens: Towards interpreting linguistic mechanisms of large language models via sparse auto-encoder. *Preprint*, arXiv: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. 2025. Saebench: A comprehensive benchmark for sparse autoencoders in language model interpretability. *Preprint*, arXiv:2503.09532.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. 2020. Concept bottleneck models. *Preprint*, arXiv:2007.04612.

Patrick Leask, Bart Bussmann, Michael T Pearce, Joseph Isaac Bloom, Curt Tigges, Noura Al Moubayed, Lee Sharkey, and Neel Nanda. 2025. Sparse autoencoders do not find canonical units of analysis. In *The Thirteenth International Conference on Learning Representations*.

Sihang Li, Wei Shi, Ziyuan Xie, Tao Liang, Guojun Ma, and Xiang Wang. 2025. Safer: Probing safety in reward models with sparse autoencoder. *Preprint*, arXiv:2507.00665.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *Preprint*, arXiv:1711.05101.

Junru Lu, Gabriele Pergola, Lin Gui, and Yulan He. 2023. Event knowledge incorporation with posterior regularization for event-centric question answering. *Preprint*, arXiv:2305.04522.

Aleksandar Makelov, Georg Lange, and Neel Nanda. 2024. Saes discover meaningful features in the ioi task. Alignment Forum.

Samuel Marks, Can Rager, Eric J. Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. 2025. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. *Preprint*, arXiv:2403.19647.

Aashiq Muhamed, Mona T. Diab, and Virginia Smith. 2025. Decoding dark matter: Specialized sparse autoencoders for interpreting rare concepts in foundation models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1604–1635, Albuquerque, New Mexico. Association for Computational Linguistics.

Anh Nguyen, Jason Yosinski, and Jeff Clune. 2019. Understanding neural networks via feature visualization: A survey. *Preprint*, arXiv: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. 2025. Steering language model refusal with sparse autoencoders. *Preprint*, arXiv: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. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Gonçalo Paulo and Nora Belrose. 2025. Sparse autoencoders trained on the same data learn different features. *Preprint*, arXiv:2501.16615.

Senthooran Rajamanoharan, Tom Lieberum, Nicolas Sonnerat, Arthur Conmy, Vikrant Varma, János Kramár, and Neel Nanda. 2024. Jumping ahead: Improving reconstruction fidelity with jumprelu sparse autoencoders. *Preprint*, arXiv:2407.14435.

Naomi Saphra and Sarah Wiegreffe. 2024. Mechanistic? *Preprint*, arXiv:2410.09087.

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. 2025. Open problems in mechanistic interpretability. *Preprint*, arXiv:2501.16496.

Dong Shu, Xuansheng Wu, Haiyan Zhao, Daking Rai, Ziyu Yao, Ninghao Liu, and Mengnan Du. 2025. A survey on sparse autoencoders: Interpreting the internal mechanisms of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 1690–1712, Suzhou, China. Association for Computational Linguistics.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Xinyuan Yan, Shusen Liu, Kowshik Thopalli, and Bei Wang. 2025. Visual exploration of feature relationships in sparse autoencoders with curated concepts. *Preprint*, arXiv:2511.06048.

Jingyuan Yang, Rongjun Li, Weixuan Wang, Ziyu Zhou, Zhiyong Feng, and Wei Peng. 2025. Lf-steering: Latent feature activation steering for enhancing semantic consistency in large language models. *Preprint*, arXiv:2501.11036.

Zhun Yang, Joohyung Lee, and Chiyoun Park. 2023. Injecting logical constraints into neural networks via straight-through estimators. *Preprint*, arXiv:2307.04347.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2025. Instruction tuning for large language models: A survey. *Preprint*, arXiv:2308.10792.

Yu Zhao, Alessio Devoto, Giwon Hong, Xiaotang Du, Aryo Pradipta Gema, Hongru Wang, Xuanli He, Kam-Fai Wong, and Pasquale Minervini. 2025. Steering knowledge selection behaviours in LLMs via SAE-based representation engineering. In *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)*, pages 5117–5136, Albuquerque, New Mexico. Association for Computational Linguistics.

Mantong Zhou, Minlie Huang, and Xiaoyan Zhu. 2019. Robust reading comprehension with linguistic constraints via posterior regularization. *Preprint*, arXiv:1911.06948.

Content of Appendix

A 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:** 8 major U.S. cities (New York, Los Angeles, Chicago, Houston, Phoenix, Philadelphia, San Antonio, San Diego)
- **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 $411 \times 461 \times 1{,}002 \times 8 \times 341 \times 101 \times 327 \approx 1.66 \times 10^{16}$ unique person profiles, ensuring minimal memorization pressure and focusing evaluation on semantic understanding rather than rote learning.

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 5 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.

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).

B.3 Optimization Hyperparameters

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"

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

Table 5: Question-answer templates for probing each semantic relation. Templates T0–T1 are used for training (in-distribution), while T2–T3 are held out for testing generalization (out-of-distribution).

what information to retrieve. We collected activations from all 12 transformer layers independently to analyze the emergence of semantic binding across network depth.

C Supervised Sparse Autoencoder

C.1 Model Architecture

Our supervised SAE extends the standard sparse autoencoder architecture with explicit relation slots and value prediction heads. The configuration is as follows:

- **Input dimension:** 768 (matching GPT-2 hidden size)
- **Total latent features:** 100,006
  - *Free slots:* 100,000 (unsupervised, sparse)
  - *Relation slots:* 6 (supervised, one per semantic relation)
- **Encoder:** Linear projection $(768 \to 100{,}006)$ with bias
- **Decoder:** Linear projection $(100{,}006 \to 768)$ with bias, initialized as pseudo-inverse of encoder
- **Value heads:** 6 independent per-relation MLPs, each $(1 \to 256 \to 50{,}257)$. Value head i maps relation slot i's activation to vocabulary logits. During training, only the value head corresponding to the ground-truth relation receives gradients. These heads are used only for diagnostic evaluation of slot informativeness, not for the main controllability experiments.

Binding Mechanism. The 6 relation slots use a softmax-based classification mechanism: $p(r \mid x) = \text{softmax}(z_{\text{rel}})$, where $z_{\text{rel}} \in \mathbb{R}^6$ contains the activations of relation slots 100,001–100,006. The alignment loss applies cross-entropy between this distribution and the one-hot ground-truth relation label, encouraging winner-take-all dynamics where the correct relation slot dominates.

C.2 Multi-Stage Training Protocol

Training proceeds in two stages to ensure stable convergence:

- **Stage 1 (Reconstruction-Only):** 50 epochs focusing solely on autoencoding quality before introducing binding constraints
- **Stage 2 (Full Supervision):** 100 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

(50 epochs), only reconstruction loss is active to stabilize the encoder-decoder. In Stage 2 (100 epochs), all six losses are jointly optimized:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{recon}}\mathcal{L}_{\text{recon}} + \lambda_{\text{sparse}}\mathcal{L}_{\text{sparse}} + \lambda_{\text{align}}\mathcal{L}_{\text{align}} + \lambda_{\text{indep}}\mathcal{L}_{\text{indep}} + \lambda_{\text{ortho}}\mathcal{L}_{\text{ortho}} + \lambda_{\text{value}}\mathcal{L}_{\text{value}} \tag{10}$$

Each component serves a specific purpose:

Reconstruction Loss.

$$\mathcal{L}_{\text{recon}} = \text{MSE}(\hat{h}, h) = \frac{1}{d}\sum_{i=1}^{d}(\hat{h}_i - h_i)^2 \tag{11}$$

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.

$$\mathcal{L}_{\text{sparse}} = \frac{1}{B \cdot n_{\text{free}}}\sum_{b=1}^{B}\sum_{j=1}^{n_{\text{free}}}|z_{b,j}| \tag{12}$$

Enforces L1 penalty on the 100,000 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.

$$\mathcal{L}_{\text{align}} = \text{CrossEntropy}(\text{softmax}(z_{\text{rel}}), y) \tag{13}$$

$$= -\frac{1}{B}\sum_{b=1}^{B}\sum_{r=1}^{6} y_{b,r} \log \frac{\exp\big(z_{b,n_{\text{free}}+r}\big)}{\sum_{r'=1}^{6}\exp\big(z_{b,n_{\text{free}}+r'}\big)}. \tag{14}$$

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.

Independence Loss.

$$\mathcal{L}_{\text{indep}} = \sum_{i \neq j}\left(\frac{1}{B}\sum_{b=1}^{B}(z_{b,i} - \bar{z}_i)(z_{b,j} - \bar{z}_j)\right)^2 \tag{15}$$

where $\bar{z}_i = \frac{1}{B}\sum_{b=1}^{B} z_{b,i}$ is the mean activation of slot i. This penalizes off-diagonal covariance among free slots, encouraging decorrelation to reduce redundancy. Disentangled features enable better interpretability and more efficient use of the latent space.

Orthogonality Loss.

$$\mathcal{L}_{\text{ortho}} = \sum_{r=1}^{6}\sum_{j=1}^{n_{\text{free}}}\left(\frac{1}{B}\sum_{b=1}^{B}(z_{b,n_{\text{free}}+r} - \bar{z}_r)(z_{b,j} - \bar{z}_j)\right)^2 \tag{16}$$

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 ensure that each relation slot carries enough information to recover the answer, we add an auxiliary value prediction objective. For each training example b, we take the activation of the ground-truth relation slot $r_b = \text{rule\_idx}_b$ and pass it through a relation-specific value head to predict the first token of the answer:

$$\mathcal{L}_{\text{value}} = \frac{1}{B}\sum_{b=1}^{B}\text{CrossEntropy}\big(\text{V}_{r_b}(z_{b,n_{\text{free}}+r_b}), t_b\big), \tag{17}$$

where t_b is the first token of the ground-truth answer and V_r is a two-layer MLP ($1 \to 256 \to 50{,}257$) that maps the activation of relation slot r to value heads' vocabulary logits.

We instantiate six separate value heads, one for each relation type, and only the head that matches the ground-truth relation r_b is applied and updated for example b. This design forces each relation slot to be predictive of the corresponding answer token and provides an additional training signal that aligns slots with answer semantics. However, these value heads are used only during training; our controllability experiments in Section 5 rely solely on the full LLM's generation after we intervene on h.

C.4 Loss Weight Selection

The loss components are weighted to balance competing objectives:

- $\lambda_{\text{recon}} = 1.0$ — Highest priority 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 (97.8% reported in Section 5.1, Table 2).
- $\lambda_{\text{indep}} = 1 \times 10^{-2}$ — Moderate decorrelation pressure among free slots. This encourages disentangled representations without interfering with primary reconstruction and binding objectives. The quadratic covariance computation scales as $O(n_{\text{free}}^2)$, so this loss is only computed when $n_{\text{free}} \leq 10{,}000$ for efficiency. In our configuration with 100,000 free slots, this term is skipped.
- $\lambda_{\text{ortho}} = 1 \times 10^{-2}$ — 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. Value heads achieve $> 90\%$ answer accuracy (91.2% reported) when predicting directly from relation slots.

C.5 Training Hyperparameters

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

- **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:

$$\text{Acc}_{\text{binding}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\left[\underset{j}{\text{argmax}}\ z_{\text{rel},j}^{(i)} = r_i\right] \quad (18)$$

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:

$$\text{Acc}_{\text{top-}k} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}[r_i \in \text{TopK}(z_{\text{rel}}^{(i)})] \quad (19)$$

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:

$$\text{Margin} = \frac{1}{N}\sum_{i=1}^{N}(z^{(i)}_{\text{rel},j_1} - z^{(i)}_{\text{rel},j_2}) \tag{20}$$

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:

$$\text{Acc}_{\text{answer}} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{1}[\hat{a}_i = a_i] \tag{21}$$

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

This metric is computed in two distinct experimental contexts:

1. **Value Head Accuracy (Binding Validation):** Measures direct answer prediction from the six trained value head MLPs without any intervention. For each question, we use $\arg\max(z_{\text{rel}})$ to identify the predicted relation slot, then apply the corresponding value head to generate an answer token. This validates that: (a) relation slots correctly bind to semantic concepts, and (b) slots contain sufficient information for answer generation. Reported accuracy: 91.2%. This is a *diagnostic metric* that does not involve the full LLM.
2. **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 ($z_i^{\text{orig}} \leftarrow 0$, $z_j^{\text{target}} \leftarrow \alpha$), we decode to obtain $\hat{h} = W_{\text{dec}} \cdot z'$ 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).

These two metrics serve complementary purposes: value head accuracy demonstrates that slots learn semantically meaningful representations, while 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:

$$\text{Diag} = \frac{1}{R}\sum_{i=1}^{R} C_{ii} \tag{22}$$

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 ≈ 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:

$$\text{MSE}_{\text{recon}} = \frac{1}{N}\sum_{i=1}^{N}\|h^{(i)} - \hat{h}^{(i)}\|^2 \tag{23}$$

where $h^{(i)}$ is the original 768-dimensional activation vector from GPT-2's residual stream and $\hat{h}^{(i)} = W_{\text{dec}} \cdot z^{(i)}$ 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 causal control over model behavior (See Section 6.5), we measure the success rate of answer swaps when intervening on SAE latents. The intervention procedure:

1. Identify the original activated relation slot i for a given question
2. Suppress the original slot: $z_i^{\text{orig}} \leftarrow 0$
3. Amplify a different target slot j: $z_j^{\text{target}} \leftarrow \alpha$
4. Decode the modified latent vector and generate an answer

We evaluate swap success across multiple amplification strengths to understand the sensitivity of the intervention:

$$\alpha \in \{0.1, 0.5, 1, 2, 5, 10, 20, 50, 100, 200, 500, 1000\} \tag{24}$$

Swap controllability is defined as:

$$\text{Swap}_\alpha = \frac{1}{M} \sum_{m=1}^{M} \mathbb{1}\big[\hat{y}_m^{\text{swap}} = r_m^\star\big], \tag{25}$$

where M is the number of swap experiments, $\hat{y}_m^{\text{swap}}$ is the relation predicted by the full language model after the intervention, and $r_m^\star$ is the ground-truth target relation for example m. The prediction $\hat{y}_m^{\text{swap}}$ is obtained by decoding the intervened SAE code z' into a modified latent vector $\hat{h} = W_{\text{dec}} z'$ and then feeding $\hat{h}$ through the remaining transformer layers, and $\mathbb{1}[\cdot]$ denotes the indicator function (1 if the condition holds, 0 otherwise). Intuitively, Swap_α measures how often amplifying a learned relation slot successfully steers the model to the corresponding target relation (*e.g.*, changing a BIRTH_DATE question to produce a BIRTH_CITY answer). As shown in Figure 5, performance is maximized at moderate amplification $\alpha \approx 2$, where Layer 6 reaches 85% swap success. Smaller values ($\alpha < 1$) provide too weak a steering signal, while very large values ($\alpha > 10$) lead to instability and degraded performance.

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

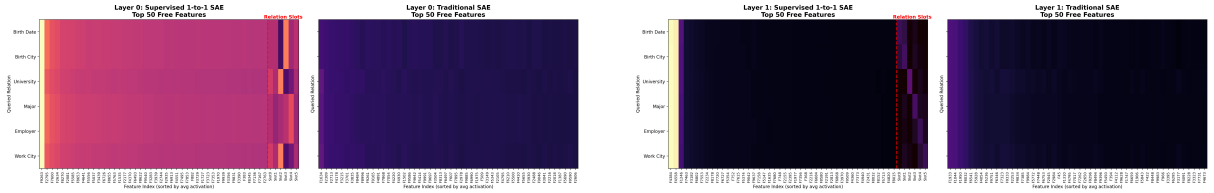
These layer-wise visualizations reveal several critical insights about the role of supervised SAE training across network depth:

Early Layers (0–3): 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 (4–8): 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.

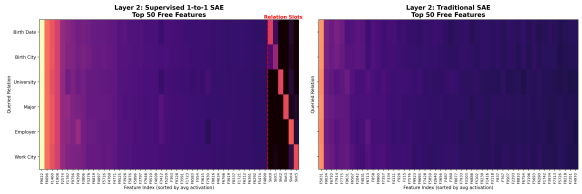
Deep Layers (6–11): Artifacts Without SAE. A particularly interesting phenomenon emerges in deeper layers without SAE post-training. Starting around Layer 6 and becoming more pronounced in Layers 9–11, the baseline (no SAE) condition exhibits strange, irregular activation patterns—potentially including 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.

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.

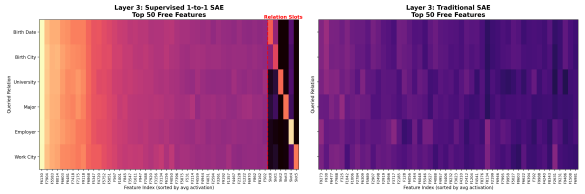


(a) Layer 0

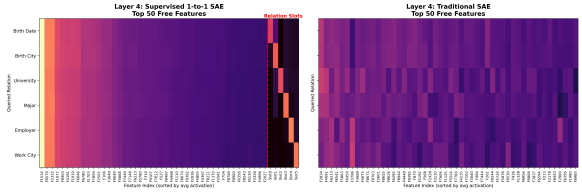
(b) Layer 1



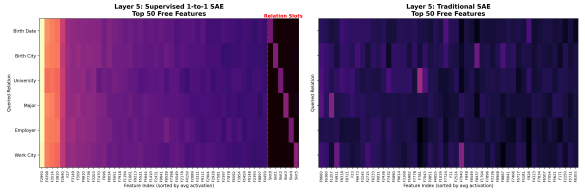
(c) Layer 2



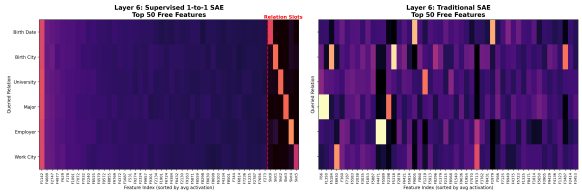
(d) Layer 3



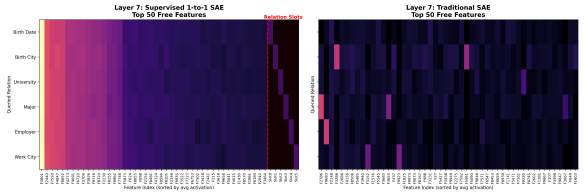
(e) Layer 4



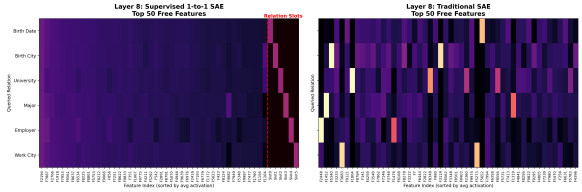
(f) Layer 5



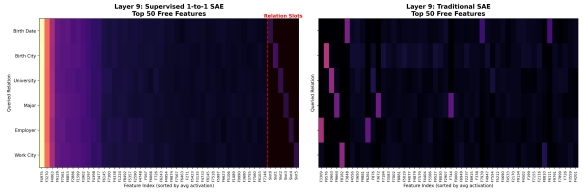
(g) Layer 6



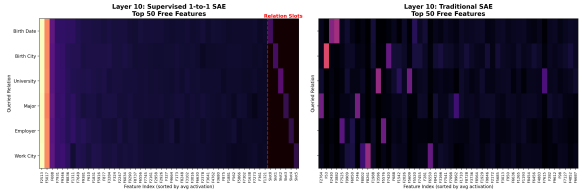
(h) Layer 7



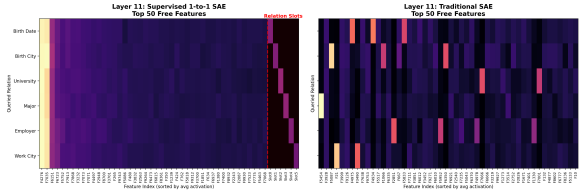
(i) Layer 8



(j) Layer 9



(k) Layer 10



(l) Layer 11

Figure 7: Layer-wise comparison of top-50 feature activations with and without SAE post-training across all 12 GPT-2 layers. Each subplot shows two conditions side-by-side. Early layers (0–3) show diffuse, entangled features regardless of SAE training. Middle layers (4–8) demonstrate the strongest benefit from SAE supervision, with clean diagonal binding emerging. Deep layers (9–11) reveal interesting artifacts: without SAE post-training, features exhibit irregular patterns and potential over-compression, while SAE-trained models maintain more structured representations. Layer 6 represents the optimal layer for semantic binding, achieving perfect diagonal accuracy with controllable relation slots.

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.