From Noise to Narrative: Tracing the Origins of Hallucinations in Transformers

Praneet Suresh[*]
Mila - Quebec AI Institute
praneet.suresh@mila.quebec

Jack Stanley[*]
Mila - Quebec AI Institute
jack.stanley@mila.quebec

Sonia Joseph
Mila - Quebec AI Institute, Meta AI
sonia.joseph@mila.quebec

Luca Scimeca
Mila - Quebec AI Institute
luca.scimeca@mila.quebec

Danilo Bzdok
Mila - Quebec AI Institute
bzdokdan@mila.quebec

Abstract

As generative AI systems become competent and democratized in science, business, and government, deeper insight into their failure modes now poses an acute need. The occasional volatility in their behavior, such as the propensity of transformer models to hallucinate, impedes trust and adoption of emerging AI solutions in high-stakes areas. In the present work, we establish how and when hallucinations arise in pre-trained transformer models through concept representations captured by sparse autoencoders, under scenarios with experimentally controlled uncertainty in the input space. Our systematic experiments reveal that the number of semantic concepts used by the transformer model grows as the input information becomes increasingly unstructured. In the face of growing uncertainty in the input space, the transformer model becomes prone to activate coherent yet input-insensitive semantic features, leading to hallucinated output. At its extreme, for pure-noise inputs, we identify a wide variety of robustly triggered and meaningful concepts in the intermediate activations of pre-trained transformer models, whose functional integrity we confirm through targeted steering. We also show that hallucinations in the output of a transformer model can be reliably predicted from the concept patterns embedded in transformer layer activations. This collection of insights on transformer internal processing mechanics has immediate consequences for aligning AI models with human values, AI safety, opening the attack surface for potential adversarial attacks, and providing a basis for automatic quantification of a model's hallucination risk.

1 Introduction

Generative AI systems are becoming always more capable, demonstrating remarkable performance on a wide range of tasks, across diverse data modalities. Yet, transformer models can at times confidently misstate facts, invent details in generated content, and engage in confabulated reasoning. These model behaviors occur even when accurate information for the task is abundantly present in the models' training data or even context window. These plausible-seeming yet incorrect outputs are

[*] Equal contribution

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

often referred to as “hallucinations”, and range from simple factual mistakes to potentially harmful and deceptive or mis-aligned behaviors [26][1][46][41][65].

While it may be trivial to quantify the error rate of a transformer model using various performance benchmarks, working towards a truly mechanistic understanding of model hallucinations, errors, and inherent biases will be unavoidable as we entrust AI systems with increasingly sensitive and consequential real-world tasks.

For the goal of dissecting the units of meaning on which transformer models internally operate, sparse autoencoders (SAEs) have imposed themselves as a widely adopted tool in the mechanistic interpretability community [12][7]. This class of autoencoders now provides the means for disentangling the opaque visceral workings of large language models (LLMs) into human understandable components. Such SAEs learn a mapping function from intermediate transformer model activations to a sparse overcomplete representational space, which has an interpretable basis by nature, that can be used to faithfully reconstruct the latent activations. Recent work has shown that human interpretable features of consequence can be identified in transformer models at scale, from toy models to commercial-scale LLMs, using SAEs [60].

Much of this interpretability research is rooted in the “linear representation hypothesis”. It posits that semantic concepts are represented as single-dimensional linear directions in a representation space derived from transformer layer activations [49][18]. While there is some evidence that certain features may be inherently multi-dimensional in SAE embedding activations [19], there is a growing body of research underscoring the validity and utility of these linearly decomposable single-dimensional features [42][51][21][18].

However, several highly important open questions remain: to what extent do these learned representations in transformer models reflect properties of the input data itself, as opposed to inherent biases or structural priors that emerged during model training? How do these internal representations behave when the transformer model is facing uncertainty due to confusing, ambiguous, or noisy inputs?

Using SAEs as a tool to probe the emergent conceptual landscape of transformer internals, we show that pre-trained transformers impose coherent, steerable conceptual structure even on pure noise or randomly shuffled inputs, with the repertoire of activated concepts expanding as input structure degrades, across both images and text. Crucially, we show that patterns of concept activations elicited by the *input prompt* reliably predict hallucinations in the transformer’s generated *output*, providing a practical signal for anticipating unfaithful generations and directly linking high-level inferred concepts to model errors. Taken together, our exploratory and interventional experiments yield three key contributions:

1. First, common pre-trained transformer models exhibit a strong form of input-insensitive inductive bias: these systems tend to impose semantic structure on inputs, tying them into learned conceptual webs, even if the model inputs are ambiguous or lack any coherent meaning.
2. Second, this skewing as information trickles through transformer processing layers exacerbates as input uncertainty increases: the more randomness we experimentally introduce in the input observations, the more persistently the transformer model is drawn towards operating on semantic units that align with familiar internal model representations. We show a demonstrable expansion of semantic concepts that are triggered, localized especially to the middle layers of the transformer model, as the coherence of input information degrades.
3. Third, the constellation of concept activations in a transformer model’s intermediate processing representations can be used to reliably predict the tendency of hallucinated or unfaithful output. This insight suggests an automatically measurable link between spurious internal feature recruitment of the input and the fidelity of the output produced by these increasingly popular deep learning systems.

2 Preliminaries

In this section, we introduce our experimental framework to probe the emergence of semantic structure in transformer models, from layer to layer, by means of SAE training and concept evaluation. To address our research aims, we employ SAEs trained on transformer activations from pure noise

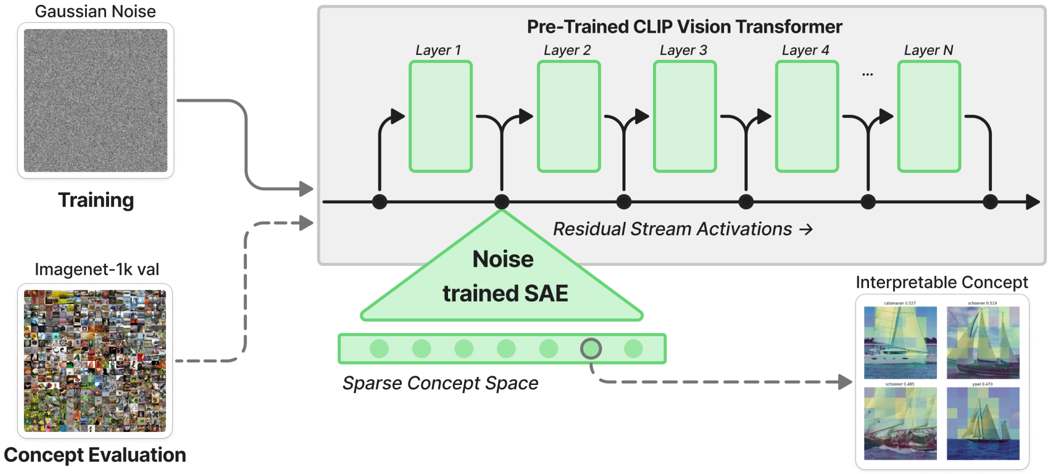


Figure 1: **Workflow to examine hallucination risk in each layer of the transformer.** To assess the extent to which transformer models infer meaningful concepts from semantically void inputs, we study the semanticity of concepts in SAEs trained on residual stream activations of 1.3 million Gaussian noise samples from a pre-trained CLIP vision transformer at each layer by probing the concepts with natural images from the ImageNet-1k validation set. Many concepts are highly interpretable and consistent, despite the SAEs only ever being exposed to pure noise residual stream activations during training. While this noise training setup is specific to our first experiment, subsequent experiments adopt the same concept evaluation approach with alternative transformer models and input modalities. See Preliminaries and Appendix A for further experimental details.

inputs, SAEs trained conventionally from scratch for smaller transformer models, and conventionally pre-trained SAEs for larger transformer models. To ensure that our results are as general and widely relevant as possible, we investigated latent activations from vision and language transformers of varying sizes (detailed in the following subsections and in Appendix A); thus covering several choices of data modality, model size, and model type.

2.1 Transformer models

For the vision transformer (ViT) [14], we study the CLIP-ViT-B/32 [53] architecture: the base variant of the vision transformer from OpenCLIP [11], pre-trained on the LAION dataset [57] with 1.4 billion image-text pairs using contrastive supervision. In the case of language transformers [62], we use Pythia-160m-deduped [4], trained on 300 billion tokens from the Pile [22]. We also use Gemma 2B-IT [59] for experiments where coherent text generation and instruction following is required. For all transformer models, the object of investigation is the activations in the residual stream: the hidden state of the deep neural network, updated by attention and multi-layer perceptron (MLP) blocks at each layer. The residual stream is the core conduit for information flow and representation refinement to which layers read from and write to in the transformer architecture [15] [16].

2.2 SAE objective

For each SAE formulation, we work with SAEs whose parameters are trained to produce the following approximation for the "residual stream activations", or individual token embeddings $\mathbf{x} \in \mathbb{R}^{d_{\text{model}}}$ for a particular layer:

$$\mathbf{x} \approx \sum_{i}^{d_{\text{SAE}}} f_i(\mathbf{x})\mathbf{d}_i + \mathbf{b}\,,$$

that is, each transformer layer's intermediate activations can be approximated as a (sparse) linear combination of unit direction vectors $\mathbf{d}_i \in \mathbb{R}^{d_{\text{model}}}$ scaled by concept activation coefficients $f_i(\mathbf{x}) \geq 0$, with $\mathbf{b} \in \mathbb{R}^{d_{\text{model}}}$ being the bias term.

Practically speaking, this SAE is a simple neural network with an encoder $F : \mathbb{R}^{d_{\text{model}}} \rightarrow \mathbb{R}^{d_{\text{SAE}}}$ projecting each transformer activation into a sparse concept space, typically of much larger dimensionality than the transformer model activation space. For instance, the ViT embedding dimensionality is $d_{\text{model}} = 768$ with the corresponding SAE concept space dimension being $d_{\text{SAE}} = 49,152$. Conversely, the learned decoder $D : \mathbb{R}^{d_{\text{SAE}}} \rightarrow \mathbb{R}^{d_{\text{model}}}$ component of the SAE aims to reconstruct the original transformer residual stream activation solely from this pattern of sparse concept activations. This linear decoder D reconstructs the activation, $\hat{\mathbf{x}} = Df(\mathbf{x}) + \mathbf{b}$, whose columns $\mathbf{d}_i$ define the concept directions.

Each SAE in this work is trained end-to-end with the following loss function:

$$\mathcal{L} = ||\mathbf{x} - \hat{\mathbf{x}}||_2^2 + \lambda ||f(\mathbf{x})||_1 \, ,$$

where the left (main) term encourages faithful reconstruction (denoted by $\hat{\mathbf{x}}$) of the residual stream activations of the transformer layer at hand, while the right (penalty) term is an L_1 regularization constraint encouraging parsimony in the concept space. The composite loss of these two terms motivates the SAE to learn a pruned yet semantically relevant mapping from dense transformer model activations to a sparse semantic concept space. Each SAE is trained independently on a single transformer layer's residual stream activations—e.g., for a 12 layer transformer, we train 12 distinct SAEs on the residual stream activations output from the corresponding layer.

2.3 SAE experimental setups

Noise trained SAEs For each layer's residual stream probe of the frozen vision transformer from the OpenCLIP ViT-B/32 model, we train a dedicated SAE on activations corresponding to 1.3 million samples of Gaussian noise. Each sample is a 3x224x224 tensor sampled independently from standard Gaussians $\mathcal{N}(0, 1)$, clipped to $\pm 3\sigma$ and min-max rescaled to $[0, 1]$ before standard ViT normalization.

SAEs trained conventionally from scratch To examine overarching principles of the thus decomposed concept spaces of pre-trained transformer models, we also train SAEs on the residual stream activations from each layer of the transformer model under investigation using conventional image and text datasets. We train SAEs on the intermediate activations of OpenCLIP ViT-B/32 from ImageNet-1k [13], and SAEs on the intermediate activations of Pythia-160m-deduped from FineWeb-Edu [50]. For each layer of these transformer models, we train a separate SAE instance; here and in all analogous experiments.

Pre-trained SAEs For larger transformer models, we use publicly available SAEs pre-trained on conventional text datasets for comparability with prior interpretability work and to allow analysis on models whose outputs are more realistic, coherent, and representative of practical use cases. In particular, we implement pre-trained Gemma 2B SAEs [5] across layers 1, 7, 11, 13, and 18.

2.4 Concept evaluation

To rule out the possibility that the features identified by the noise setting SAE are mere artifacts of the inductive bias of the SAE, we subject each feature in a randomly sampled subset of the SAE features to two complementary, model-agnostic metrics, designed for vision:

1. Semantic purity: To quantify the unequivocality with which a particular feature f_i is present, we collect the top-k ($k = 16$) images that maximize the mean activation of f_i across the sequence of image patches, obtain their CLIP text embeddings (label) and compute the mean cosine similarity (-1.0 to +1.0) of the text embeddings. High semantic purity (close to +1.0) indicates that the images cluster around a single coherent semantic theme. A concept is "pure" if the top-k ($k = 16$ in our case) maximally activating images for that concept have an average cosine similarity of greater than 0.75 across their semantic label embeddings.
2. Steerability: Steerability is the capability of flipping a neutral image's semantic label to that of the concept, by clamping the concept's activation to an artificially high value. To try to "hijack" a particular feature f_i in the transformer model, we add the scaled SAE decoder direction $\alpha \cdot \mathbf{d}_i$ of that feature to the residual stream activations. The scalar α is chosen via a grid search over a pre-defined range of values. We then run the transformer model on feature-naive "neutral" image inputs that originally do not activate f_i. If this single vector

addition to a given transformer model layer makes the model's top-1 prediction switch to the feature's own majority label (the concept identified from its images that activate maximally) we consider f_i as "steerable". In our case, conservatively, if $n = 32$ out of 32 images from a randomly sampled batch of neutral images flip to the concept's label after clamping, then this concept is said to be "steerable".

2.5 Hallucination

Evaluation of model hallucination is generally recognized to be challenging [26] [41]. As a straightforward empirical test, we prompt Gemma 2B-IT to summarize a long-form text, and employ the purpose-built and widely used HHEM-2.1 model [3] to provide a continuous hallucination score from 0.0 (no hallucination) to 1.0 (maximum hallucination) for the summary based on its faithfulness to the original text. To directly link hallucinations to our SAE-identified concepts, we fit a partial least squares (PLS) regression layer by layer to predict the hallucination score of the transformer output based on the concept activations of the input prompt. PLS is a linear model that enables interpretable links between concepts and hallucination behavior. To estimate the hallucination score, h, we fit:

$$\hat{h} = \mathbf{X}_{\max}\mathbf{W}(\mathbf{P}^T\mathbf{W})^{-1}\mathbf{Q}^T \, ,$$

where $\mathbf{X}_{\max} \in \mathbb{R}^{n_{\text{sample}} \times d_{\text{SAE}}}$ is calculated by taking the maximum concept activations across input tokens for a source, $\mathbf{W} \in \mathbb{R}^{d_{\text{SAE}} \times n_{\text{comp}}}$ is the learned projection matrix into the PLS latent space, $\mathbf{P} \in \mathbb{R}^{d_{\text{SAE}} \times n_{\text{comp}}}$ are the concept loadings, and $\mathbf{Q} \in \mathbb{R}^{1 \times n_{\text{comp}}}$ are the hallucination score loadings. More details can be found in Appendix E.

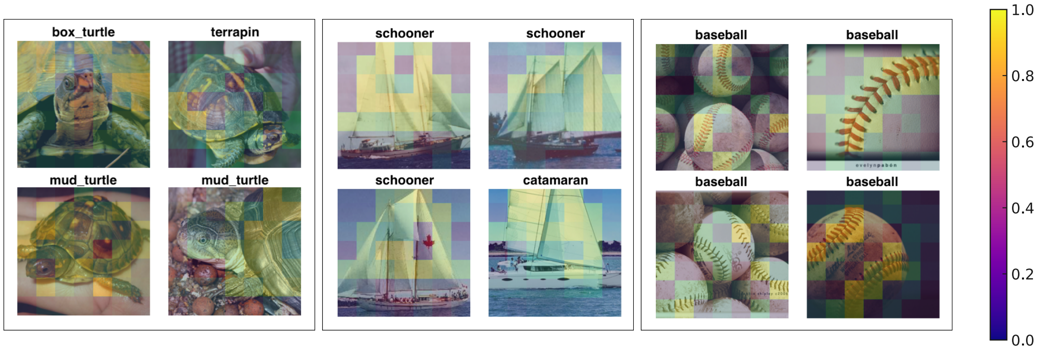


Figure 2: **Semantic concepts are reliably invoked by vision transformer layer 9 activations from pure noise.** Each panel shows the top 4 images among 50,000 candidate images from the ImageNet-1k validation set that most activated a particular semantic concept. These semantic concepts are defined by an SAE trained on pure noise activations from the residual stream at layer 9 of a CLIP vision transformer. Above each image we report the corresponding semantic label for that image. Patch colors indicate the individual patch activation strengths within that image for a given semantic concept (yellow = more activation of the concept). See Appendix B for additional concept examples.

3 Transformers represent semantic concepts for pure noise inputs

To investigate the emergence of semantic structure when pretrained transformers are presented with structurally incoherent inputs, we train SAEs on the residual stream activations from pure noise inputs to the CLIP vision transformer. If the pre-trained vision transformer is able to interpret semantic structure from inputs that by construction contain no semantic signal, the ensuing identified semantic concepts could be said to reveal the model's ingrained conceptual biases (Figure 1).

After training one SAE instance per layer on the residual stream activations for 1.3 million Gaussian noise samples, we probe a randomly sampled subset of the noise SAE-identified concepts using 50,000 example images from the ImageNet-1k validation set. To evaluate the interpretability of these concepts, we obtain the top-k ($k = 16$ in this case) maximally activating images (mean concept activation across all patches) for each concept (Figure 2). Additional concept examples from our noise trained SAEs are presented in Appendix B. We then computed the semantic purity of the concept (cf. Preliminaries) as a proxy for its interpretability, computed by the cosine similarity of the semantic CLIP labels from the top 16 maximally activating evaluation images.

We define a feature as interpretable if its semantic purity is at least 0.75. Using this strict threshold, we observe many high purity features, each corresponding to a coherent high level concept, despite the SAE being trained solely on Gaussian noise activations and never seeing naturalistic image activations during training. We also observe that a subset of these highly interpretable concepts are steerable (cf. Preliminaries): injecting the concept-related activation vector into the residual stream as the transformer model is processing a neutral image can shift the model's semantic label prediction toward the label associated with the concept of interest (Figure 3).

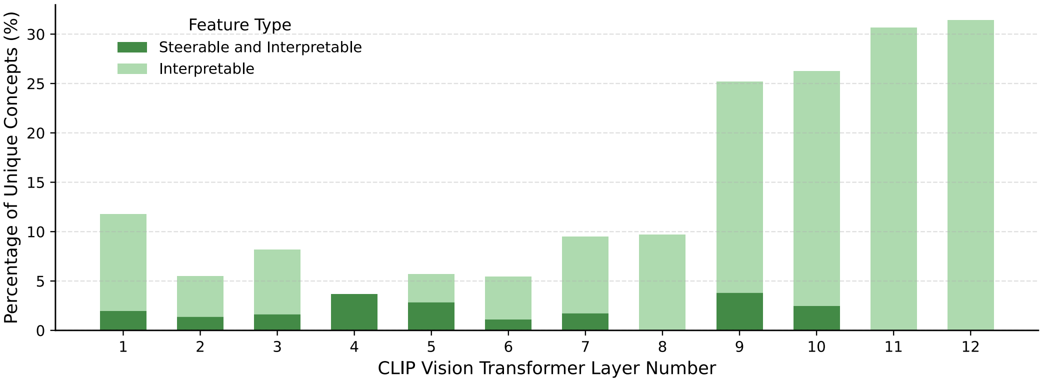
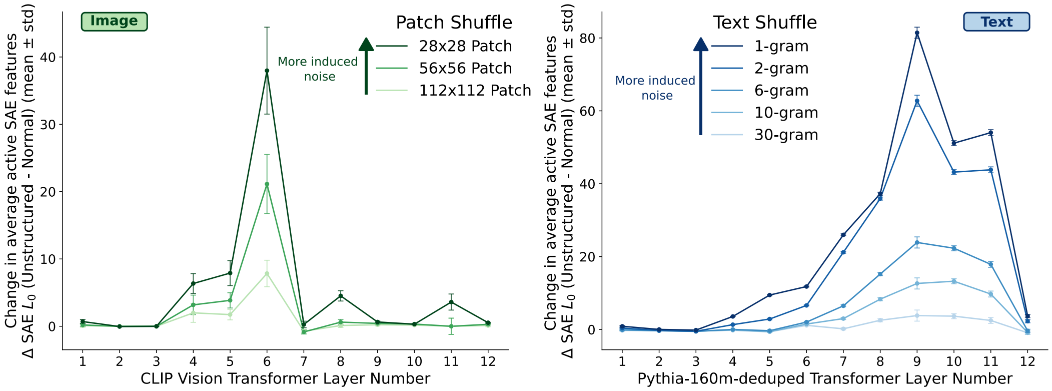


Figure 3: **Transformers fed with noise inputs lead to neuron activations with detectable and controllable semantic structure in many layers.** We measure the interpretability of a concept by the semantic similarity of the labels of the top 16 images that maximally activate that concept. As a more stringent test, we also measure a concept's steerability by the ability of that concept to causally induce its own class label when added to the residual stream of neutral input images. We find that a very large portion of our noise-derived semantic concepts are highly interpretable, and a non-negligible number of these concepts are even steerable, particularly in the early and middle layers. We report the percentage of unique concepts across our noise-trained SAE meeting the aforementioned interpretability and steerability thresholds.

As a further test of concept robustness and layer-wise structuring, we train different noise SAE instances with varying seeds and hyperparameters on the same transformer layer activations. A consistent core of robust concepts [47] emerges across comparable layers. Comparing two sets of noise-setting SAEs reveals a distinctive three-phase pattern in concept overlap, offering insight into how the vision transformer structures information (Appendix C, Figure 16). The first three layers share 35% of their concepts, because the residual stream here still reflects the statistical structure of the raw input, and the space of the viable concepts is highly constrained given the minimal inductive bias at the onset of processing. Conceptual overlap drops sharply to 14% at layers 5-6, marking a stochastic exploration phase within the vision transformer's latent space, exploring a much broader semantic hypothesis space reducing the probability that randomly sampled features from both sets of SAEs will have a large overlap. In deeper layers of the vision transformer model, overlap rebounds to 25% onto a more stable set of high-level features. See Appendix C for further details on these overlapping concepts. In effect, the model settles on the most explanatory subset of concepts for the task, collapsing the expanded mid layer space back into a shared semantic scaffold. Taken together, this three phase pattern indicates that the vision transformer incrementally restructures noise-based activations into coherent task-relevant concepts, and that these emergent representations are sufficiently robust to be discovered by noise-setting SAEs trained with different random initializations.

Grounding our evaluation with the ImageNet-1k validation set gives us a strict, reproducible baseline. Since the concept space is limited by the samples in the validation set, any feature aligned with concepts beyond ImageNet's dictionary is treated as unclassified, making our interpretability and steerability counts understated. A more diverse evaluation set, with images supporting a wider array of ground-truth concepts, would likely show an increase in both of these metrics. Regardless, we have demonstrated that transformer models ascribe distinct, coherent, and causally actionable concepts to inputs that, by construction, have no semantic content.



(a)

(b)

Figure 4: **Increasing uncertainty in vision or text inputs elicits more semantic structure in mid-layers of transformers.** The average number of SAE concepts identified (L0) increases dramatically with increasing input perturbation. We report the average change in L0 from baseline, corresponding to the number of SAE concepts with non-zero activations, across (a) patch-shuffled image activations and (b) n-gram-shuffled text activations for each transformer layer. Error bars show one standard deviation. Smaller patches and lower n-gram count induce greater input uncertainty for images and text, respectively. For both modalities, the L0 difference between natural inputs and perturbed inputs peaks in the middle layers, and increases with increasing levels of deliberate scrambling of transformer input information.

4 More unstructured transformer inputs invoke more semantic concepts

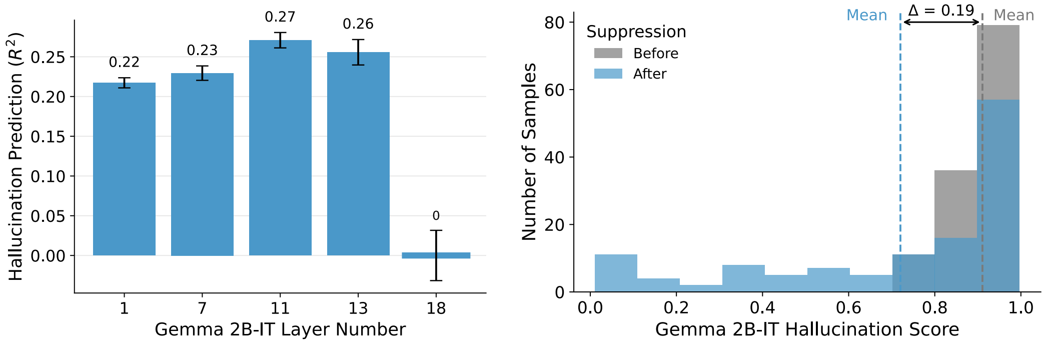
To get a better grip on how the transformer models progressively mold the conceptual structure embedded in the input as information flows through the processing layers, we train one conventional SAE instance on each transformer layer independently. These per-layer SAEs are trained on transformer activations from natural images/text, not noise. We provide these normal input activations as a baseline, and then compare the differences in the SAE concept space for varying levels of experimentally induced ambiguity in the input examples. In the case of image inputs to the vision transformer, we randomly shuffle small patches to break the spatial relationship between adjacent parts of the image in natural images (28x28, 56x56, and 112x112 pixel patches shuffled per image), with smaller patches leading to more ambiguous inputs. To this end, for text inputs to the language transformer, we randomly shuffle sets of consecutive words in varying n-gram sizes (single, 2, 6, 10, and 30 word sections), with lower n introducing higher uncertainty in the input space. The transformer layer's ensuing residual stream activations corresponding to these inputs are then fed into the appropriate SAEs for each layer.

A useful macro-level metric for overall SAE concept discovery is the SAE L0: the average number of non-zero concepts across the input sequence activated through transformer processing over the entire dataset. In the case of our image SAEs, remarkably, we notice a steep increase in the average number of concepts activated for randomly shuffled inputs compared to the baseline, peaking at layer 6 with 38 additional concepts activated on average (Figure 4a) for the smallest 28x28 pixel patch size. This pattern holds in the language domain as well. The number of concepts identified per input above the baseline peaks at layer 9 with 81 extra concepts triggered, on average (Figure 4b) for the single word shuffled text. Furthermore, we observe that the degree of input perturbation strongly influences the magnitude of this effect. Input activations with finer-grained shuffling (e.g. single word level or smaller image patches) result in significantly higher L0 count values than input activations shuffled in larger chunks (e.g. 30-word sections or large image patches). See Appendix D for full tables of layer-wise results (Tables 6-7).

These results, converging across different data modalities, are suggestive of the middle layers of pre-trained transformer models playing a more influential role in mediating the model's internal conceptual instantiation in response to degraded input structure. One might expect transformer models to intuit fewer concepts in shuffled inputs; instead, we find that ambiguity drives an expansion in concept usage, revealing an unanticipated property of how these transformer models organize and impose structure on input processing cascades.

5 Input-related internal concept activations predict output hallucinations

We have seen how pre-trained transformer models, when confronted with noisy, uncertain, and ambiguous inputs, appear to traverse the landscape of possible concepts more widely, attempting to impose structure that may not actually be present. Next, we aim to directly link this transformer model-imposed structure in the concept space of the inputs to hallucination propensity with respect to the model-generated outputs. To this end, we first instructed Gemma 2B-IT to produce a generated summary for 1,006 articles from the Vectara hallucination leaderboard [24], employing the purpose-built HHEM-2.1 rating model to give each source/summary pair a real-valued hallucination score, ranging from 0.0 (no hallucination) to 1.0 (most hallucinated). Simultaneously, the transformer's residual stream activations of the original article text were projected into the sparse concept space formed by the pre-trained Gemma 2B SAEs. With these concept vectors in hand, we fit a 4-component PLS on the maximum concept activations across tokens in a sample, attempting to predict the hallucination score of the Gemma 2B-IT-generated text summary.



(a) (b)

Figure 5: **Transformer layer activations can be used to directly predict risk of hallucinated model output.** (a) Hallucination score prediction for the task of faithful summarization of 1,006 source articles. We compare Gemma 2B-IT-generated summaries against the ground truth source articles. We use the sparse SAE concept activations, derived from Gemma 2B-IT residual stream activations, as input to a PLS regression model, predicting the hallucination score for each example. We report 10-fold cross-validated coefficient of determination (R^2) on unseen examples, with error bars showing one standard deviation. (b) Suppressing the top 10 SAE concepts in Layer 11's residual stream, identified by the PLS model to be the primary drivers of hallucination, significantly reduces mean hallucination scores across the top quartile most hallucinated examples ($n = 252$). We show a histogram of hallucination scores before (grey) and after (blue) suppression: many examples report significant reductions in hallucination, with a mean score drop of 0.19 in this subset (dashed lines).

Indeed, we find that hallucinations present in the Gemma 2B-IT-generated summaries can be robustly predicted, in unseen text examples, solely from the sparse concept vectors derived from the layer activations of the source prompt using the SAEs. The most accurate predictions are observed for the layer 11 residual stream SAE concepts with a coefficient of determination of 0.271 ± 0.010 (out of sample, 10-fold cross validation) for the continuous hallucination scores. For binary classification predicting above versus below the median hallucination score, we reach a mean accuracy of $73.0\% \pm 5.3\%$ across 10 cross validation folds. Transformer layer 18 SAE source concepts do not allow us to predict the continuous summary hallucination scores above chance level (Figure 5a). In Appendix E we also show that our method outperforms existing hallucination detection methods on several additional short form generation and multiple choice hallucination benchmark datasets, despite the fact that we only take into account concepts in the input prompt (Tables 9-12).

Based on this independent mode of investigation, our results reinforce our previous findings (cf. above) that it is especially the intermediate transformer model layers where concept exploration and risk of “conceptual wandering" occur. According to the present analyses, later transformer layers do not appear to participate in this process to a large extent.

We finally explore the practical implications of these findings. Since PLS is a linear model, we can easily reverse-engineer specific key concepts in each transformer layer that have the most influence on the hallucination score predictions. This is because each PLS input dimension corresponds to a distinct SAE-derived concept activation. We select the top 10 key hallucination-associated concepts in layer 11 by variance importance in projection (VIP). With these 10 concepts in hand, we simply suppress them in the residual stream of layer 11 for the input prompt by setting their SAE concept activations to 0, with the SAE decoder yielding hallucination-suppressed residual stream activations. These hallucination-suppressed activations are then swapped in instead of the actual vector of residual stream activations in Gemma 2B-IT’s forward pass. Intriguingly, for the top quartile most hallucinated examples in our dataset (252 samples in total), suppressing just these 10 features (out of a total of 16,384) leads to a decrease in the mean hallucination score of 0.19 for this subset (Figure 5b), from 0.91 to 0.72 after suppression. Further details are provided in Appendix E.

This constellation of results shows a clear link between concept space activations of the input prompt and the tendency of the transformer model to hallucinate its generated content. More than this, we present a primitive mechanism to tamp down this tendency to hallucinate.

6 Related Work

Hallucination and perturbed inputs The occurrence of hallucinations in generated content is a well-documented issue, persisting across language models regardless of scale or capability [26] [65] [61]. These hallucinations refer to model-generated outputs that are syntactically plausible but factually incorrect or ungrounded to the input prompt. There have been many attempts to benchmark this phenomenon across various tasks [20][24][2][36]. There have been relatively fewer attempts to explain the mechanisms within transformers that underlie these behaviors. [64] attempts to explain (and control) hallucinations based on syntactic errors in particular components of the transformer architecture such as specific attention heads or MLPs. [27] looks at hallucination through the lens of output token dynamics. [32] studies model self-evaluation, and [33] rigorously shows that well-calibrated models will always hallucinate on certain “arbitrary” facts at a baseline rate. These approaches largely employ custom-made datasets and work on the level of single token probabilities and token-wise syntactic dynamics. Our approach links the phenomenon of hallucination directly to interpretable, high-level concepts, and as far as we know is the first study to present a unifying mechanism across both images and text. Our approach also bears some similarities to a large body of research on neural network input perturbation and adversarial examples, where small, often imperceptible changes to the input can lead to erroneous model outputs [58][9][45][63][28][23]. In our work, we introduce highly perturbed inputs as an experimental tool to test the interaction of the actual semantic content of the inputs and the information imposed on the inputs by the transformer.

Sparse autoencoders Early word embedding approaches, such as word2vec [44] and GloVe [52], demonstrated the power of linearly composable embedding spaces for capturing high-level semantic relationships in natural language [43]. More recently, this result has been formalized through the linear representation hypothesis [49], which suggests that many meaningful features of input data can be disentangled and inspected via linear transformations [18][6][17]. Leveraging this insight, sparse autoencoders (SAEs) have proven to be particularly effective tools for breaking down dense high dimensional transformer representations into an overcomplete basis of linear pieces that can be independently studied [12][7][40]. SAEs have surfaced coherent and even manipulable concepts from transformer models of various sizes [60][34]; recent work has extended these techniques to vision transformers with similar success [30][29][25]. There has also been much focus on improving the intelligibility and relevance of SAE-identified features through various alternative training regimes and architectures [21][54][8]. Our experiments leverage SAE-disentangled concepts to localize transformer model behaviors such as hallucinations and conceptual drift, with results that rely solely on the presence of high-level concepts rather than specific SAEs, datasets, or even data modalities. Our approaches are likely to gain strength as interpretability techniques advance.

7 Conclusion

We here introduce an objective framework for detecting and profiling structural biases and failure modes in common pre-trained transformer models. Our approach scales organically to transformer models of any size. Crucially, it allows for dataset-independent assessments—eliminating reliance on arbitrary benchmark data subsets and subjective observer-dependent judgements such as highlighting a handful of specific semantic concepts or zooming in on small functional circuits in billion-parameter transformer models. By systematic probing of internal representations as they trickle through the transformer architecture following simple random noise and other controlled input perturbations, we show how to compute a continuous degree of hallucination risk, even on a layer-by-layer basis. We surface presupposed semantic concepts of interest, study the trajectory of their propagation throughout the deep neural network, and examine the nature of how experimentally perturbed inputs become internally represented through fully quantitative, automatically computed measures. In doing so, we reveal important ways in which transformer models are prone to "conceptual wandering", invoking rich semantic structure in inputs with pure noise or degraded semantic information content. Such frameworks may shift interpretability efforts to see deeper into transformer models from qualitative judgements and hand-picked examples to a pipelinable methodology that generalizes across multiple data modalities, expansive model architectures, and experimental setups. Ultimately, our work lays a cornerstone for both real-time monitoring and principled intervention of transformer model behavior, supporting the safe and responsible deployment of emergent AI abilities to the wider society.

Limitations While our experiments span a broad range of model sizes and modalities, it remains to be seen how exactly these findings may manifest in the largest frontier models on the order of 10-100+ billion parameters. Nonetheless, the consistency of observed behaviors across architectures and scales in this work suggests strong potential for generalization. As well, interesting future work would be to explore the interplay of hallucination with increased sequence length and sampling temperature. We would expect increased hallucination and concept activation in these settings. We do not claim that SAEs are perfect oracles for interpretability; rather, our study builds upon the observation that SAEs have been shown repeatedly (in this work and many other works) to extract many useful semantic concepts from transformer activations. Indeed, we have shown here that these concepts can be used to directly predict relevant behavior of model-generated content.

References

[1] Jean-Baptiste Alayrac et al. *Flamingo: a Visual Language Model for Few-Shot Learning*. arXiv:2204.14198 [cs]. Nov. 2022. DOI: 10.48550/arXiv.2204.14198.

[2] Yejin Bang et al. *HalluLens: LLM Hallucination Benchmark*. arXiv:2504.17550 [cs]. Apr. 2025. DOI: 10.48550/arXiv.2504.17550.

[3] Forrest Bao et al. *HHEM-2.1-open*. 2024. DOI: 10.57967/hf/3240.

[4] Stella Biderman et al. *Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling*. arXiv:2304.01373 [cs]. May 2023. DOI: 10.48550/arXiv.2304.01373.

[5] Joseph Bloom. "Open source sparse autoencoders for all residual stream layers of gpt2 small". In: *AI alignment forum*. 2024, p. 24.

[6] Tolga Bolukbasi et al. *Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings*. arXiv:1607.06520 [cs]. July 2016. DOI: 10.48550/arXiv.1607.06520.

[7] Trenton Bricken et al. "Towards monosemanticity: Decomposing language models with dictionary learning". In: *Transformer Circuits Thread* (2023).

[8] Bart Bussmann et al. *Learning Multi-Level Features with Matryoshka Sparse Autoencoders*. arXiv:2503.17547 [cs]. Mar. 2025. DOI: 10.48550/arXiv.2503.17547.

[9] Nicholas Carlini and David Wagner. *Towards Evaluating the Robustness of Neural Networks*. arXiv:1608.04644 [cs]. Mar. 2017. DOI: 10.48550/arXiv.1608.04644.

[10] Chao Chen et al. *INSIDE: LLMs' Internal States Retain the Power of Hallucination Detection*. arXiv:2402.03744 [cs]. Oct. 2024. DOI: 10.48550/arXiv.2402.03744.

[11] Mehdi Cherti et al. "Reproducible scaling laws for contrastive language-image learning". In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. arXiv:2212.07143 [cs]. June 2023, pp. 2818–2829. DOI: 10.1109/CVPR52729.2023.00276.

[12] Hoagy Cunningham et al. *Sparse Autoencoders Find Highly Interpretable Features in Language Models*. arXiv:2309.08600 [cs]. Oct. 2023. DOI: 10.48550/arXiv.2309.08600.

[13] Jia Deng et al. "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. ISSN: 1063-6919. June 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[14] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. arXiv:2010.11929 [cs]. June 2021. DOI: 10.48550/arXiv.2010.11929.

[15] Nelson Elhage, Robert Lasenby, and Christopher Olah. "Privileged bases in the transformer residual stream". In: *Transformer Circuits Thread* (2023), p. 24.

[16] Nelson Elhage et al. "A mathematical framework for transformer circuits". In: *Transformer Circuits Thread* (2021).

[17] Nelson Elhage et al. *Toy Models of Superposition*. arXiv:2209.10652 [cs]. Sept. 2022. DOI: 10.48550/arXiv.2209.10652.

[18] Nelson Elhage et al. "Toy models of superposition". In: *Transformer Circuits Thread* (2022).

[19] Joshua Engels et al. *Not All Language Model Features Are One-Dimensionally Linear*. arXiv:2405.14860 [cs]. Feb. 2025. DOI: 10.48550/arXiv.2405.14860.

[20] Sebastian Farquhar et al. "Detecting hallucinations in large language models using semantic entropy". en. In: *Nature* 630.8017 (June 2024). Publisher: Nature Publishing Group, pp. 625–630. ISSN: 1476-4687. DOI: 10.1038/s41586-024-07421-0.

[21] Leo Gao et al. *Scaling and evaluating sparse autoencoders*. arXiv:2406.04093 [cs]. June 2024. DOI: 10.48550/arXiv.2406.04093.

[22] Leo Gao et al. *The Pile: An 800GB Dataset of Diverse Text for Language Modeling*. arXiv:2101.00027 [cs]. Dec. 2020. DOI: 10.48550/arXiv.2101.00027.

[23] Dan Hendrycks and Thomas Dietterich. *Benchmarking Neural Network Robustness to Common Corruptions and Perturbations*. arXiv:1903.12261 [cs]. Mar. 2019. DOI: 10.48550/arXiv.1903.12261.

[24] Simon Hughes, Minseok Bae, and Miaoran Li. "Vectara hallucination leaderboard". In: *Vectara, Inc* 466 (2023).

[25] hugofry. "Towards Multimodal Interpretability: Learning Sparse Interpretable Features in Vision Transformers". en. In: (Apr. 2024).

[26] Ziwei Ji et al. “Survey of Hallucination in Natural Language Generation”. In: *ACM Computing Surveys* 55.12 (Dec. 2023). arXiv:2202.03629 [cs], pp. 1–38. ISSN: 0360-0300, 1557-7341. DOI: 10.1145/3571730.

[27] Che Jiang et al. *On Large Language Models’ Hallucination with Regard to Known Facts*. arXiv:2403.20009 [cs]. Oct. 2024. DOI: 10.48550/arXiv.2403.20009.

[28] Di Jin et al. *Is BERT Really Robust? A Strong Baseline for Natural Language Attack on Text Classification and Entailment*. arXiv:1907.11932 [cs]. Apr. 2020. DOI: 10.48550/arXiv.1907.11932.

[29] Sonia Joseph et al. *Prisma: An Open Source Toolkit for Mechanistic Interpretability in Vision and Video*. arXiv:2504.19475 [cs]. Apr. 2025. DOI: 10.48550/arXiv.2504.19475.

[30] Sonia Joseph et al. *Steering CLIP’s vision transformer with sparse autoencoders*. arXiv:2504.08729 [cs] version: 1. Apr. 2025. DOI: 10.48550/arXiv.2504.08729.

[31] Mandar Joshi et al. *TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension*. arXiv:1705.03551 [cs]. May 2017. DOI: 10.48550/arXiv.1705.03551.

[32] Saurav Kadavath et al. *Language Models (Mostly) Know What They Know*. arXiv:2207.05221 [cs]. Nov. 2022. DOI: 10.48550/arXiv.2207.05221.

[33] Adam Tauman Kalai and Santosh S. Vempala. *Calibrated Language Models Must Hallucinate*. arXiv:2311.14648 [cs]. Mar. 2024. DOI: 10.48550/arXiv.2311.14648.

[34] Tom Lieberum et al. *Gemma Scope: Open Sparse Autoencoders Everywhere All At Once on Gemma 2*. arXiv:2408.05147 [cs]. Aug. 2024. DOI: 10.48550/arXiv.2408.05147.

[35] Chin-Yew Lin. “ROUGE: A Package for Automatic Evaluation of Summaries”. In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 74–81.

[36] Stephanie Lin, Jacob Hilton, and Owain Evans. *TruthfulQA: Measuring How Models Mimic Human Falsehoods*. arXiv:2109.07958 [cs]. May 2022. DOI: 10.48550/arXiv.2109.07958.

[37] Zi Lin, Jeremiah Zhe Liu, and Jingbo Shang. “Towards Collaborative Neural-Symbolic Graph Semantic Parsing via Uncertainty”. en. In: *Findings of the Association for Computational Linguistics: ACL 2022*. Dublin, Ireland: Association for Computational Linguistics, 2022, pp. 4160–4173. DOI: 10.18653/v1/2022.findings-acl.328.

[38] Weitang Liu et al. *Energy-based Out-of-distribution Detection*. arXiv:2010.03759 [cs]. Apr. 2021. DOI: 10.48550/arXiv.2010.03759.

[39] Andrey Malinin and Mark Gales. *Uncertainty Estimation in Autoregressive Structured Prediction*. arXiv:2002.07650 [stat]. Feb. 2021. DOI: 10.48550/arXiv.2002.07650.

[40] Samuel Marks et al. *Sparse Feature Circuits: Discovering and Editing Interpretable Causal Graphs in Language Models*. arXiv:2403.19647 [cs]. Mar. 2025. DOI: 10.48550/arXiv.2403.19647.

[41] Joshua Maynez et al. “On Faithfulness and Factuality in Abstractive Summarization”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by Dan Jurafsky et al. Online: Association for Computational Linguistics, July 2020, pp. 1906–1919. DOI: 10.18653/v1/2020.acl-main.173.

[42] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. “Linguistic Regularities in Continuous Space Word Representations”. In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Lucy Vanderwende, Hal Daumé III, and Katrin Kirchhoff. Atlanta, Georgia: Association for Computational Linguistics, June 2013, pp. 746–751.

[43] Tomas Mikolov et al. “Distributed Representations of Words and Phrases and their Compositionality”. In: *Advances in Neural Information Processing Systems*. Vol. 26. Curran Associates, Inc., 2013.

[44] Tomas Mikolov et al. *Efficient Estimation of Word Representations in Vector Space*. arXiv:1301.3781 [cs]. Sept. 2013. DOI: 10.48550/arXiv.1301.3781.

[45] Seyed-Mohsen Moosavi-Dezfooli et al. *Universal adversarial perturbations*. arXiv:1610.08401 [cs]. Mar. 2017. DOI: 10.48550/arXiv.1610.08401.

[46] Long Ouyang et al. *Training language models to follow instructions with human feedback*. arXiv:2203.02155 [cs]. Mar. 2022. DOI: 10.48550/arXiv.2203.02155.

[47] Isabel Papadimitriou et al. *Interpreting the Linear Structure of Vision-language Model Embedding Spaces*. arXiv:2504.11695 [cs]. Apr. 2025. DOI: 10.48550/arXiv.2504.11695.

[48] Kishore Papineni et al. "Bleu: a Method for Automatic Evaluation of Machine Translation". In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Ed. by Pierre Isabelle, Eugene Charniak, and Dekang Lin. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, July 2002, pp. 311–318. DOI: 10.3115/1073083.1073135.

[49] Kiho Park, Yo Joong Choe, and Victor Veitch. *The Linear Representation Hypothesis and the Geometry of Large Language Models*. arXiv:2311.03658 [cs]. July 2024. DOI: 10.48550/arXiv.2311.03658.

[50] Guilherme Penedo et al. *The FineWeb Datasets: Decanting the Web for the Finest Text Data at Scale*. arXiv:2406.17557 [cs]. Oct. 2024. DOI: 10.48550/arXiv.2406.17557.

[51] Jeffrey Pennington, Richard Socher, and Christopher Manning. "GloVe: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Alessandro Moschitti, Bo Pang, and Walter Daelemans. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. DOI: 10.3115/v1/D14-1162.

[52] Jeffrey Pennington, Richard Socher, and Christopher Manning. "GloVe: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Alessandro Moschitti, Bo Pang, and Walter Daelemans. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. DOI: 10.3115/v1/D14-1162.

[53] Alec Radford et al. *Learning Transferable Visual Models From Natural Language Supervision*. arXiv:2103.00020 [cs]. Feb. 2021. DOI: 10.48550/arXiv.2103.00020.

[54] Senthooran Rajamanoharan et al. *Improving Dictionary Learning with Gated Sparse Autoencoders*. arXiv:2404.16014 [cs]. Apr. 2024. DOI: 10.48550/arXiv.2404.16014.

[55] Siva Reddy, Danqi Chen, and Christopher D. Manning. *CoQA: A Conversational Question Answering Challenge*. arXiv:1808.07042 [cs]. Mar. 2019. DOI: 10.48550/arXiv.1808.07042.

[56] Jie Ren et al. *Out-of-Distribution Detection and Selective Generation for Conditional Language Models*. arXiv:2209.15558 [cs]. Mar. 2023. DOI: 10.48550/arXiv.2209.15558.

[57] Christoph Schuhmann et al. "LAION-5B: An open large-scale dataset for training next generation image-text models". en. In: *Advances in Neural Information Processing Systems* 35 (Dec. 2022), pp. 25278–25294.

[58] Christian Szegedy et al. *Intriguing properties of neural networks*. arXiv:1312.6199 [cs]. Feb. 2014. DOI: 10.48550/arXiv.1312.6199.

[59] Gemma Team et al. *Gemma: Open Models Based on Gemini Research and Technology*. arXiv:2403.08295 [cs]. Apr. 2024. DOI: 10.48550/arXiv.2403.08295.

[60] Adly Templeton et al. "Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet". In: *Transformer Circuits Thread* (2024).

[61] S. M. Towhidul Islam Tonmoy et al. *A Comprehensive Survey of Hallucination Mitigation Techniques in Large Language Models*. arXiv:2401.01313 [cs]. Jan. 2024. DOI: 10.48550/arXiv.2401.01313.

[62] Ashish Vaswani et al. *Attention Is All You Need*. arXiv:1706.03762 [cs]. Aug. 2023. DOI: 10.48550/arXiv.1706.03762.

[63] Eric Wallace et al. *Universal Adversarial Triggers for Attacking and Analyzing NLP*. arXiv:1908.07125 [cs]. Jan. 2021. DOI: 10.48550/arXiv.1908.07125.

[64] Lei Yu et al. "Mechanistic Understanding and Mitigation of Language Model Non-Factual Hallucinations". In: *Findings of the Association for Computational Linguistics: EMNLP 2024*. Ed. by Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen. Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 7943–7956. DOI: 10.18653/v1/2024.findings-emnlp.466.

[65] Yue Zhang et al. *Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models*. arXiv:2309.01219 [cs]. Sept. 2023. DOI: 10.48550/arXiv.2309.01219.

A SAE hyperparameters and training metrics

In Tables 1-3, we report the hyperparmeters used for training SAEs from scratch in this work. In Tables 4-5, we report evaluation metrics for SAEs that we train from scratch in this work.

Table 1: Hyperparameters for OpenCLIP ViT-B/32 SAEs trained on residual stream activations of Gaussian noise inputs. Note that the number of sampled Gaussian noise inputs matches the size of the ImageNet-1k training set, to ensure a fair comparison. Training time was approximately 35 minutes on a single NVIDIA L40S GPU.

Layer	learning rate	batch size	L_1 coef.	d_{SAE}	d_{model}	# training images
1	3e-3	4096	2e-6	49,152	768	1.3M
2	1e-3	4096	4e-6	49,152	768	1.3M
3	7e-3	4096	2e-6	49,152	768	1.3M
4	2e-3	4096	2e-6	49,152	768	1.3M
5	9e-4	4096	2e-6	49,152	768	1.3M
6	3e-3	4096	1e-6	49,152	768	1.3M
7	7e-3	4096	1e-6	49,152	768	1.3M
8	6e-3	4096	1e-6	49,152	768	1.3M
9	1e-2	4096	3e-6	49,152	768	1.3M
10	3e-4	4096	1e-5	49,152	768	1.3M
11	1e-4	4096	1e-6	49,152	768	1.3M
12	1e-4	4096	1e-6	49,152	768	1.3M

Table 2: Hyperparameters for OpenCLIP ViT-B/32 SAEs trained on residual stream activations of natural ImageNet-1k images. Training time was approximately 45 minutes on a single NVIDIA L40S GPU.

Layer	learning rate	batch size	L_1 coef.	d_{SAE}	d_{model}	# training images
1	6e-04	4096	1e-1	49,152	768	1.3M
2	1e-02	4096	3e-1	49,152	768	1.3M
3	2e-03	4096	4e-1	49,152	768	1.3M
4	1e-04	4096	2e-8	49,152	768	1.3M
5	2e-04	4096	3e-8	49,152	768	1.3M
6	3e-04	4096	1e-8	49,152	768	1.3M
7	2e-03	4096	1e-8	49,152	768	1.3M
8	7e-04	4096	1e-8	49,152	768	1.3M
9	1e-02	4096	3e-8	49,152	768	1.3M
10	1e-02	4096	9e-8	49,152	768	1.3M
11	1e-02	4096	4e-10	49,152	768	1.3M
12	1e-02	4096	3e-7	49,152	768	1.3M

Table 3: Hyperparameters for Pythia-160m-deduped SAEs trained on residual stream activations of natural FineWeb-Edu text. Training time was approximately 4 hours on a single NVIDIA L40S GPU.

Layer	learning rate	batch size	L_1 coef.	d_{SAE}	d_{model}	# training tokens
1 — 11	5e-5	2048	5	24,576	768	204.8M
12	5e-5	2048	2	24,576	768	204.8M

Table 4: Evaluation metrics for OpenCLIP ViT-B/32 SAEs trained on residual stream activations of natural ImageNet-1k images.

Layer	L0	explained variance	recon. cos sim.	% alive features
1	989.19	0.99	0.99	100.00
2	757.83	0.99	0.99	45.39
3	1007.89	0.99	0.99	97.93
4	935.06	0.99	0.99	100.00
5	965.15	0.99	0.99	100.00
6	966.38	0.99	0.99	100.00
7	1006.62	0.99	0.99	99.97
8	984.19	0.99	0.99	100.00
9	965.12	0.99	1.00	92.37
10	854.92	0.99	1.00	85.43
11	1141.99	0.99	1.00	100.00
12	829.09	0.99	1.00	55.71

Table 5: Evaluation metrics for Pythia-160m-deduped SAEs trained on residual stream activations of natural FineWeb-Edu text.

Layer	L0	explained variance	recon. cos sim.	% alive features
1	29.50	0.92	0.97	100.00
2	16.18	0.85	0.95	100.00
3	13.25	0.81	0.93	100.00
4	21.41	0.84	0.91	98.84
5	22.31	0.77	0.90	97.35
6	31.98	0.72	0.92	89.58
7	45.89	0.79	0.92	87.28
8	65.53	0.82	0.93	87.80
9	67.55	0.81	0.93	80.48
10	49.12	0.75	0.92	84.20
11	31.57	0.70	0.92	93.95
12	11.03	0.71	0.98	22.34

B Additional noise trained SAE concept examples

We present additional examples of concepts identified by the SAEs trained on Gaussian noise activations from the residual stream of the OpenCLIP ViT-B/32, at various layers (Figures 6-15). Evaluation is carried out using 50,000 images from the ImageNet-1k validation set. Early layers largely identify concepts associated with low-level visual features and repeating patterns. Later layers identify very coherent and high-level concepts. These figures extend the results presented in Figure 2.

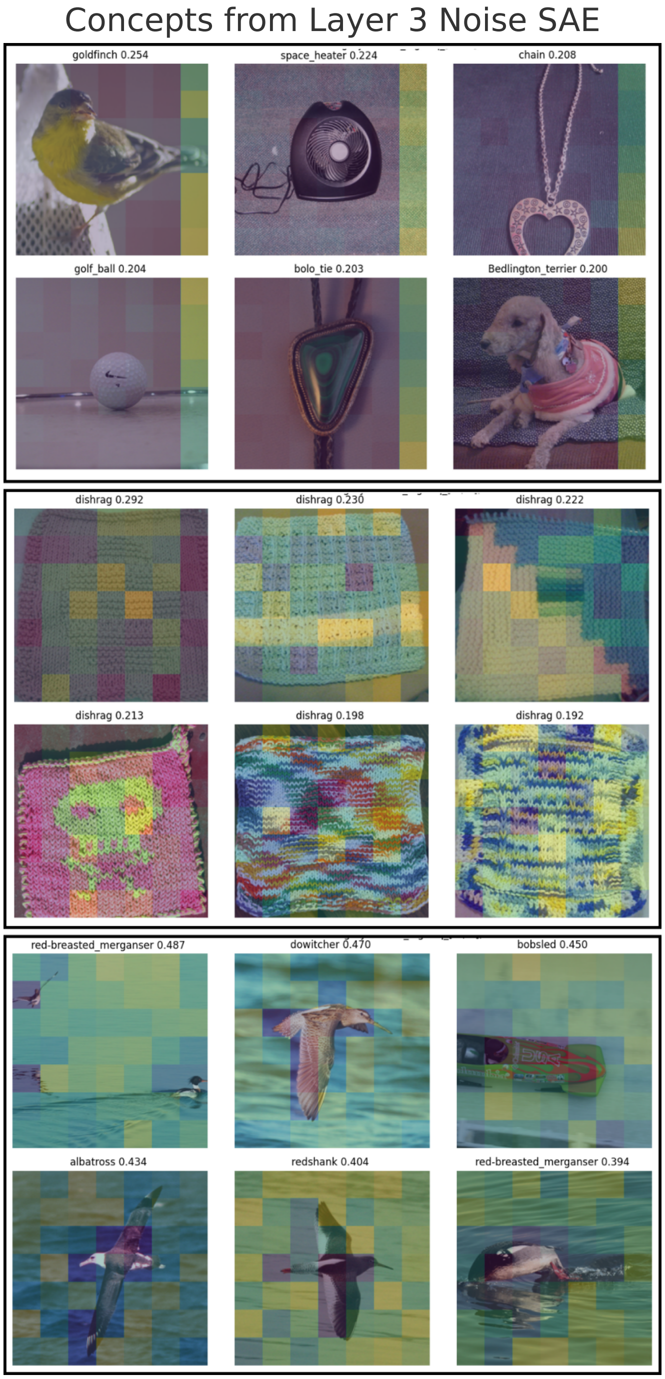


Figure 6: **Maximally activating images for three features from an SAE trained on Layer 3 activations of a ViT with Gaussian noise input**. These features, also representative of early transformer layers (Layers 1 and 2), include: (i) a positional feature that consistently activates for the final patch token across rows, (ii) a texture-sensitive feature that activates for highly repetitive, noise-like patterns resembling dishrags, and (iii) a feature capturing water-like backgrounds with birds in the foreground.

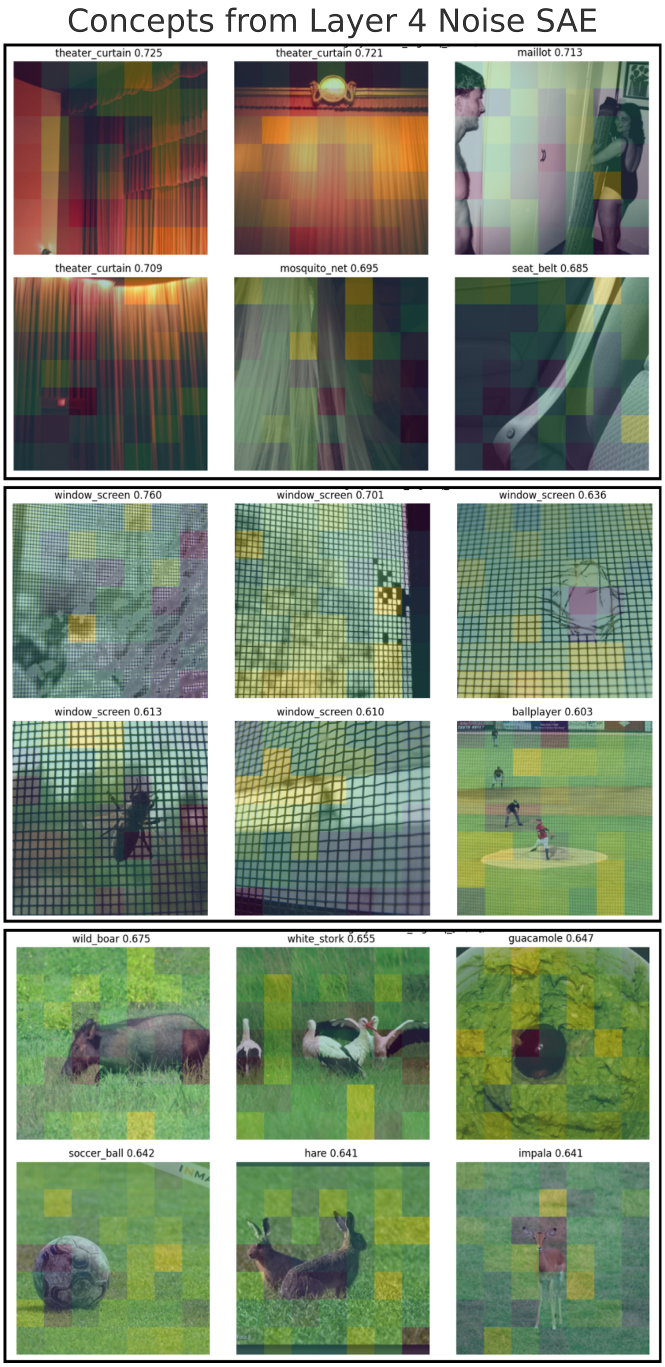


Figure 7: **Maximally activating images for three features from an SAE trained on Layer 4 activations of a ViT with Gaussian noise input**. These features remain grounded in low-level visual patterns but display more variation compared to earlier layers, without forming coherent semantic concepts: i) a feature that activates for vertically folded fabrics such as theater curtains and mosquito nets, likely driven by repetitive linear structures; ii) a feature that responds to mesh-covered textures, such as window shades with overlaid nets, producing a noisy visual appearance; iii) a feature that activates for grassy textures, possibly due to fine-grained spatial repetition.

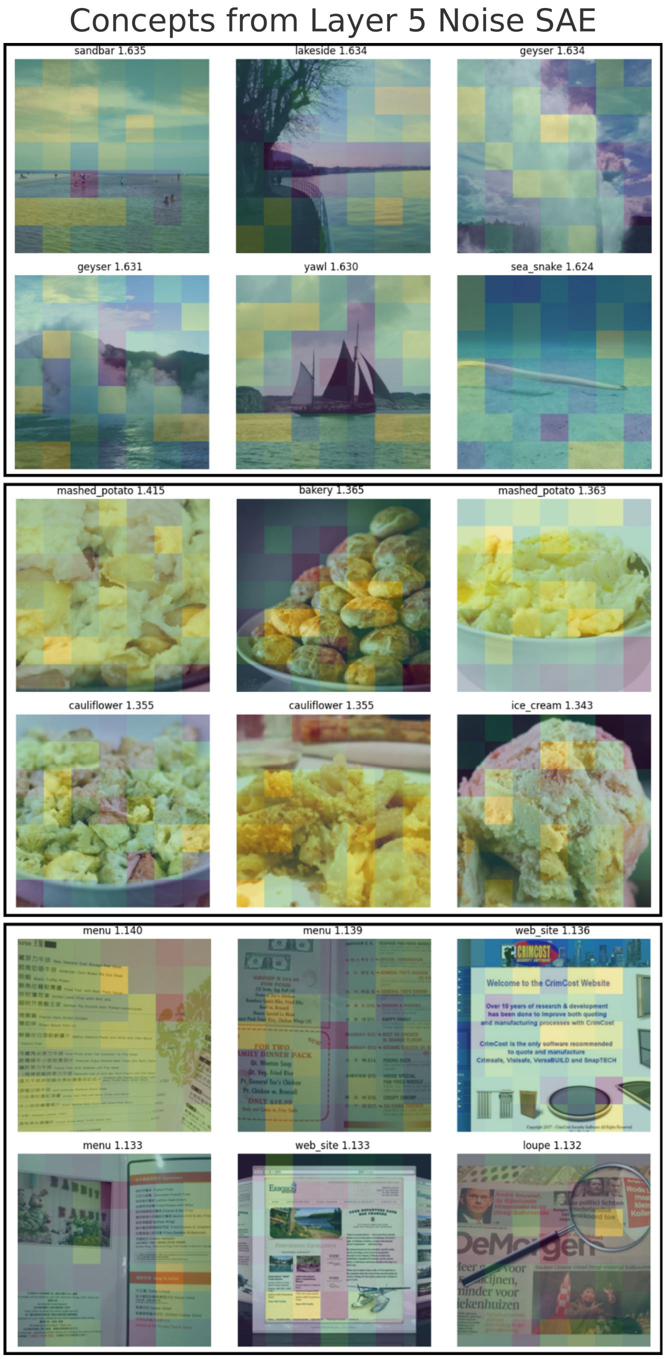


Figure 8: **Maximally activating images for three features from an SAE trained on Layer 5 activations of a ViT with Gaussian noise input**. At this layer, semantic structure begins to emerge, marking a departure from purely textural or position-based features: i) a feature that activates for natural scenes containing clouds and water bodies, indicating sensitivity to environmental layouts; ii) a feature that responds to food items with soft, clumped textures—such as mashed potatoes, ice cream, and cauliflower—highlighting texture-based abstraction; iii) a feature that activates for the presence of printed text, capturing more symbolic and structured content.

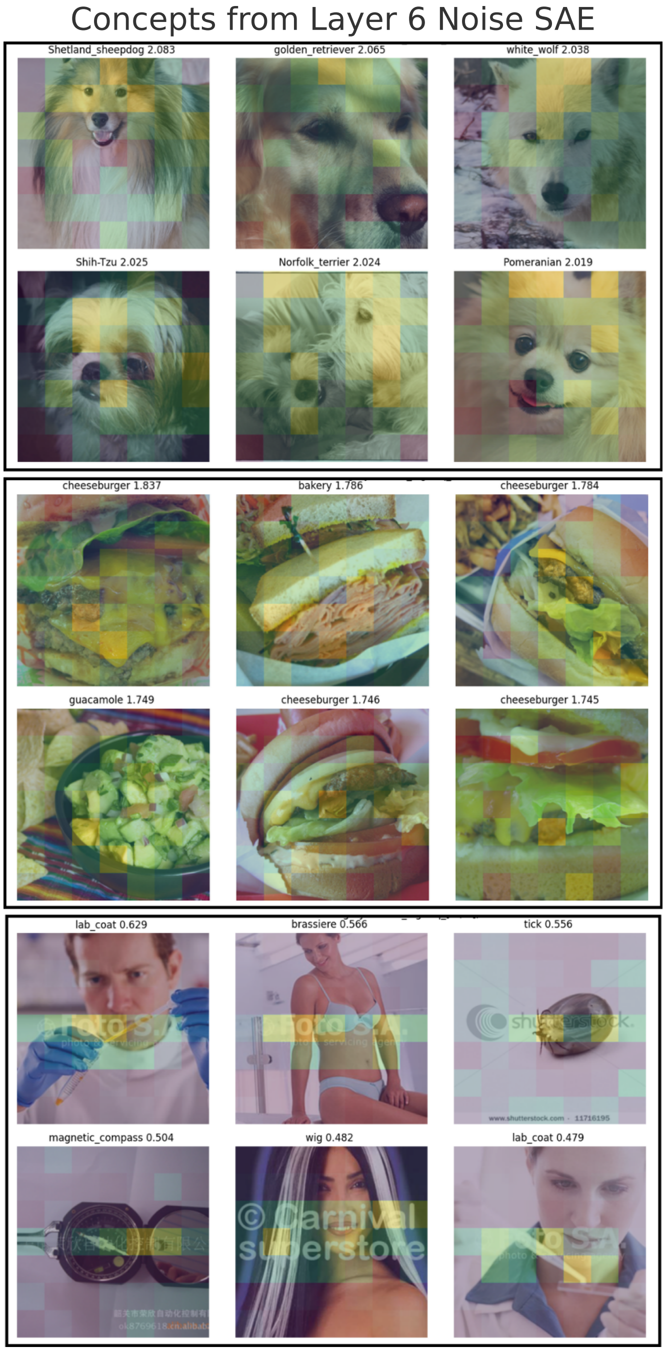


Figure 9: **Maximally activating images for three features from an SAE trained on Layer 6 activations of a ViT with Gaussian noise input**. At this layer, we have features with clear semantics: i) a feature that activates for furry dogs; ii) a feature that activates for food items such as cheeseburgers; iii) a feature that activates for the watermarks on images.

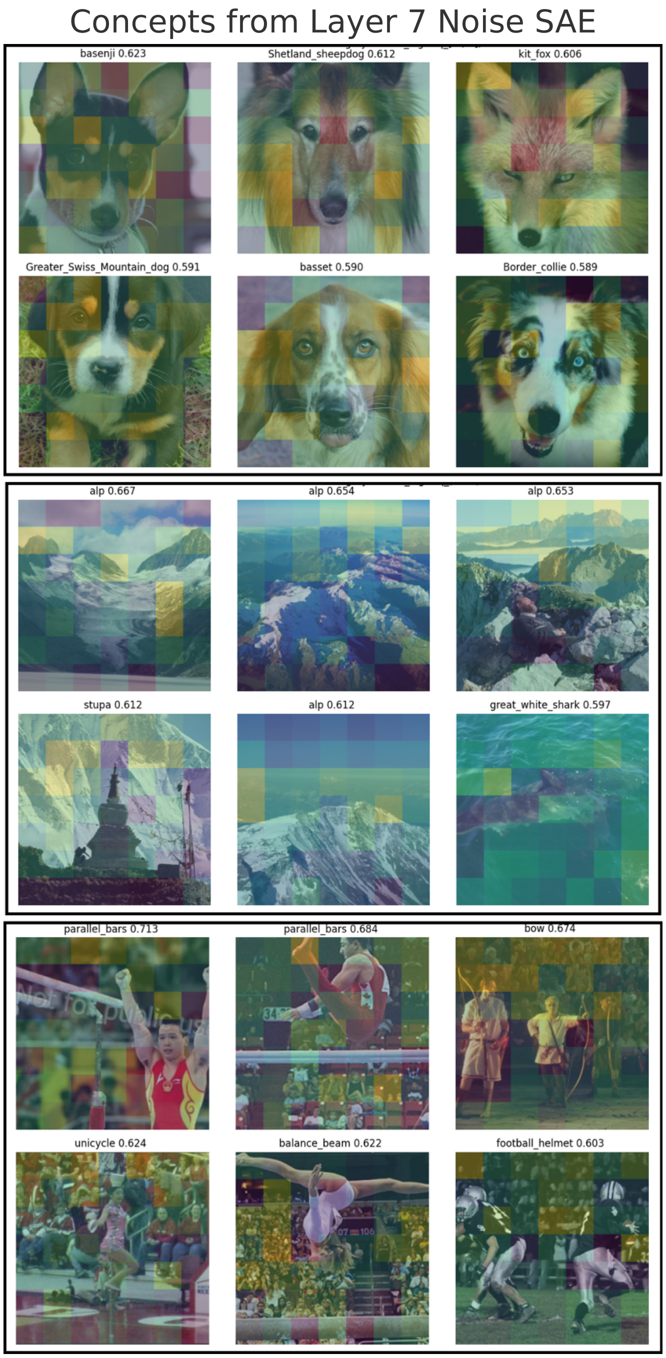


Figure 10: **Maximally activating images for three features from an SAE trained on Layer 7 activations of a ViT with Gaussian noise input**. The features at this depth increasingly correspond to distinct semantic categories and complex scene elements: i) a feature that activates for dogs with upright posture and distinct facial structure, reflecting refined animal categorization; ii) a feature that responds to natural landscapes featuring prominent vertical structures such as mountain peaks, suggesting emerging scene-level abstraction; iii) a feature that activates for human crowds or densely populated scenes, particularly those with small, background-scale human figures.

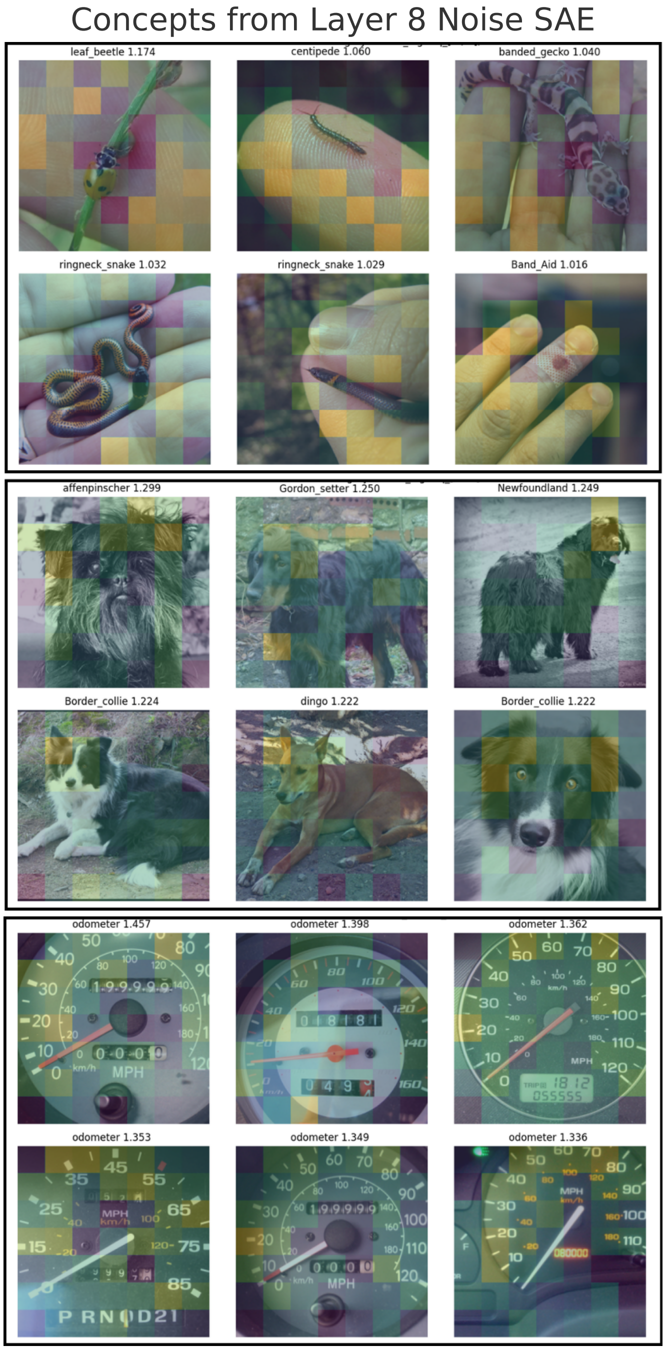


Figure 11: **Maximally activating images for three features from an SAE trained on Layer 8 activations of a ViT with Gaussian noise input**. At this stage, features begin to consolidate around coherent semantic clusters that span both foreground entities and background textures: i) a feature that activates for human fingers, in the context of them being used to hold some organic objects; ii) a feature that activates for dog ears, showing refined selectivity for specific facial features; iii) a feature that strongly activates for odometers.

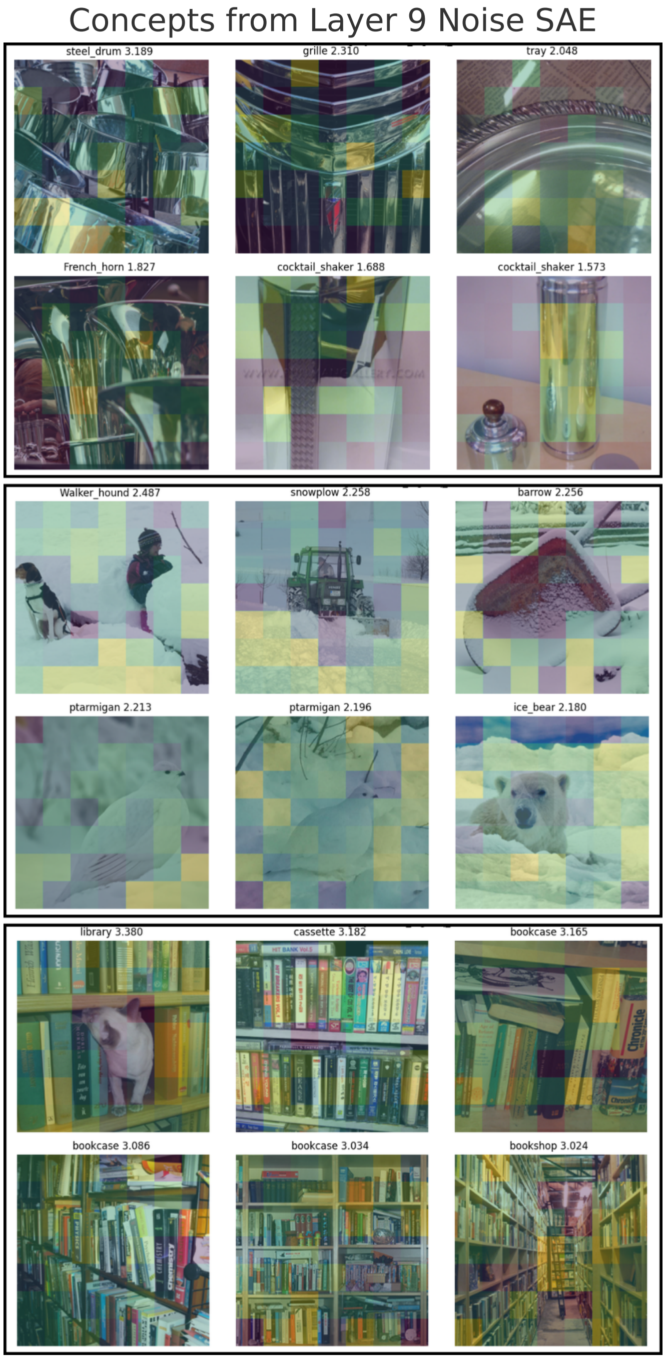


Figure 12: **Maximally activating images for three features from an SAE trained on Layer 9 activations of a ViT with Gaussian noise input**. These features reveal increasingly abstracted and high-level regularities in structured visual scenes: i) a feature that activates for glossy or metallic surfaces with high reflectance, capturing regularities in material properties and lighting conditions; ii) a feature that activates for scenes with snow-covered elements; iii) a feature that activates for bookshelves or library aisles.

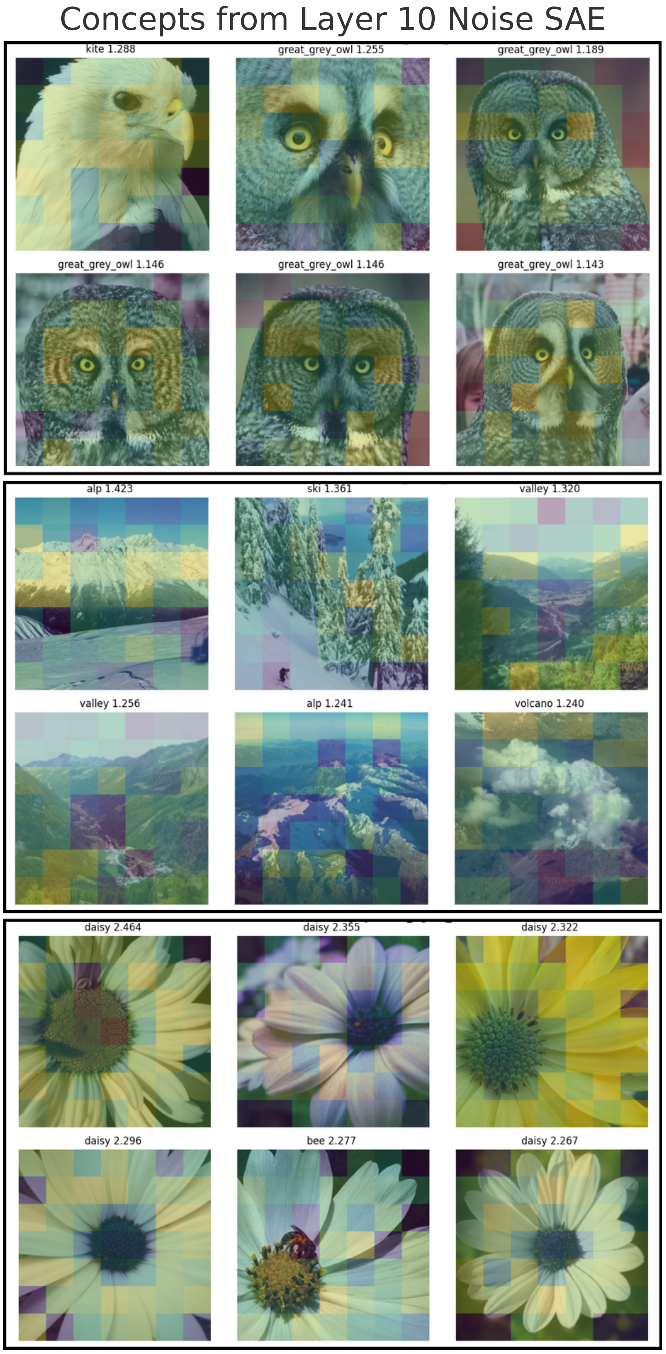


Figure 13: **Maximally activating images for three features from an SAE trained on Layer 10 activations of a ViT with Gaussian noise input**. These features demonstrate increasing semantic clarity and category specificity: i) a feature that activates for close-up views of birds of prey and owls, capturing high-level animal features such as eyes and beaks; ii) a feature that activates for elevated natural landscapes, including snowy peaks, alpine valleys, and volcanoes, emphasizing large-scale geographic structures; iii) a feature that activates for floral structures—especially daisy-like flowers.

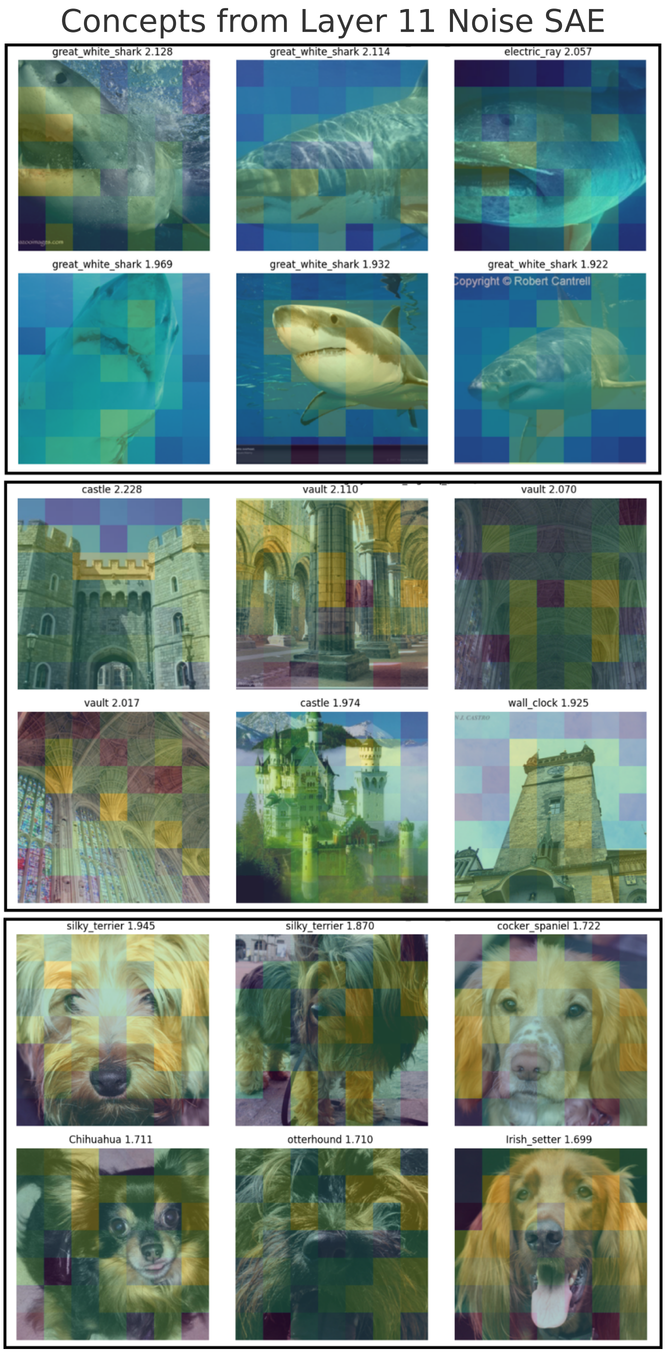


Figure 14: **Maximally activating images for three features from an SAE trained on Layer 11 activations of a ViT with Gaussian noise input**. These features exhibit advanced semantic grouping and fine-grained category emergence: i) a feature that activates for sharks, in different perspectives underwater; ii) a feature that activates for historical architecture and vaulted interiors; iii) a feature that activates for small to medium-sized furry dog breeds.

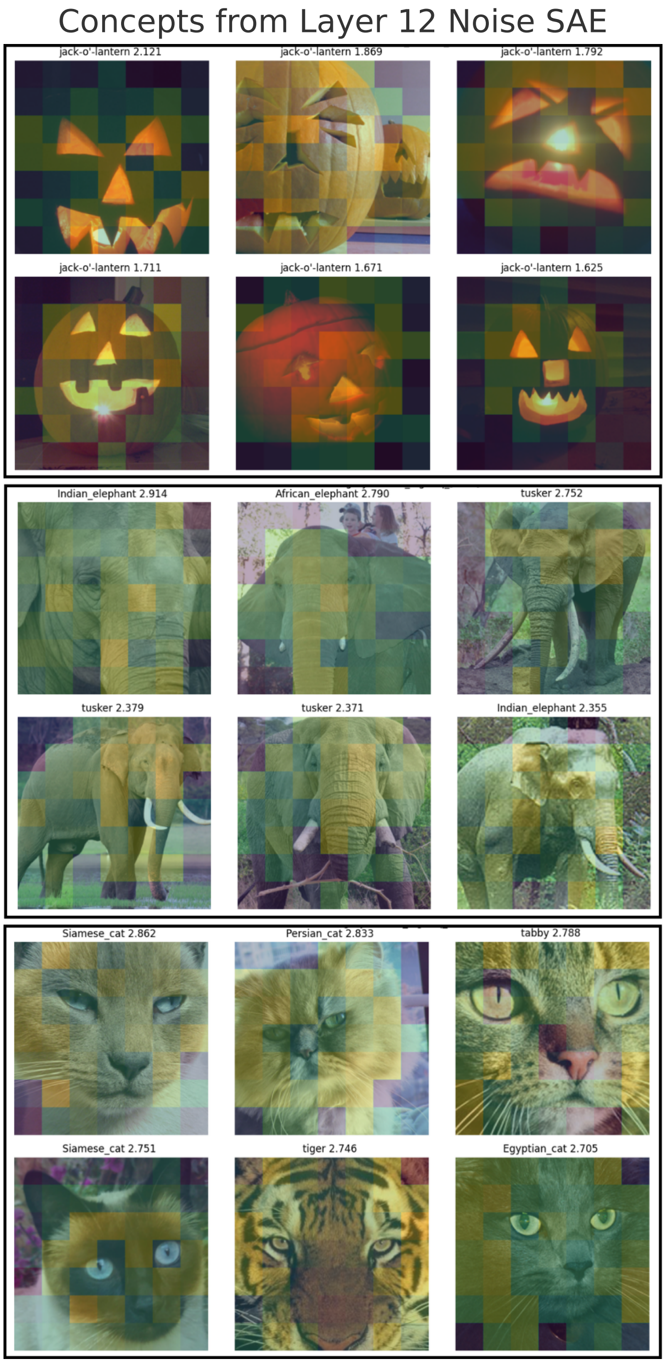


Figure 15: **Maximally activating images for three features from an SAE trained on Layer 12 activations of a transformer with Gaussian noise input**. At this deepest layer, we observe strong category-level alignment and rich semantic specificity: i) a feature that activates for jack-o'-lanterns, capturing lighting, shape, and facial carve patterns across varied contexts; ii) a feature that consistently activates for elephants, invariant to pose and background, indicating robust object identity abstraction; iii) a feature that activates for cat faces, spanning multiple breeds and subspecies, reflecting high-resolution encoding of facial features and textures.

C Noise trained SAE concept overlap with different initializations

We train two sets of noise SAE instances across all layers with different initialization seeds on the vision transformer residual stream activations derived from Gaussian noise inputs. We then compute the Jaccard index on a randomly sampled feature subset, to estimate the amount of concepts shared between these two sets of SAEs. The Jaccard index is calculated by dividing the number of overlapping concepts by the total number of concepts identified by both SAEs in the subset. For ease of reference, we superimpose the difference in L0 observed for natural image inputs to the L0 observed for maximally shuffled image inputs onto Figure 16 (see Figure 4a for reference). Interestingly, the Jaccard index between the two noise SAEs trained with different random seeds has an inverse pattern to the change in L0 from baseline to unstructured inputs. This provides further evidence that the vision transformer engages in "conceptual wandering" in its middle layers, since the noise SAEs trained from different initializations infer disjoint sets of concepts in the middle layer activations of Gaussian noise. We see this distinct three phase pattern: (i) early layers identify consistent, low-level features, (ii) middle layers widely explore the concept space and impose structure that may not be present in the inputs, and (iii) later layers crystallize the explored concepts into more consistent outputs.

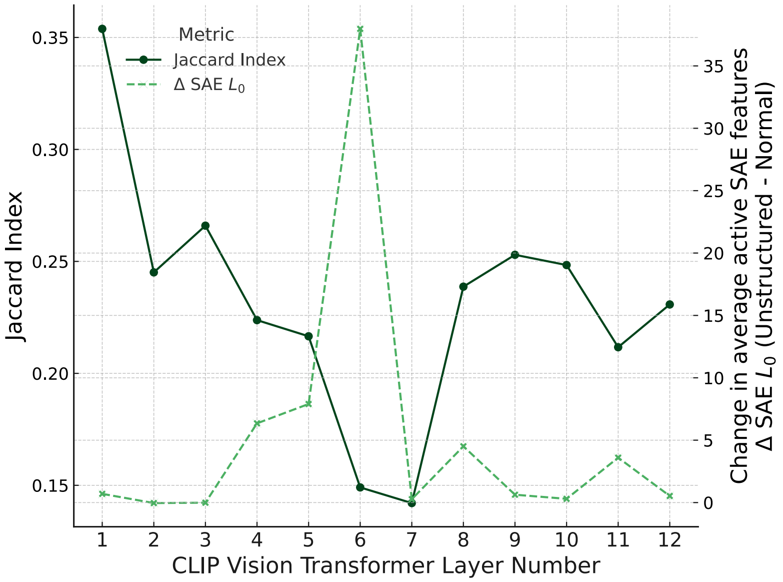


Figure 16: **Conceptual overlap dips in the middle layers of the vision transformer**. We plot the Jaccard index between the concepts identified by SAEs trained on Gaussian noise for two different random initializations (left Y-axis). We have also super-imposed the change in L0 plot for the 28x28 patch shuffled images from Figure 4a for reference (right Y-axis). There is significantly more concept overlap in the early and later layers, which declines in the middle layers. This is an inverse relationship to the increase in L0 observed for patch shuffled images.

D Layer-wise L0 for unstructured inputs

We provide mean L0s for varying levels of unstructured inputs, for image and text SAEs. Here we present raw mean L0s and standard deviations in Tables 6-7, as opposed to Figure 4 which shows the change in L0 from the normal image/text baseline.

Table 6: Mean layer-wise L0 for SAE trained on OpenCLIP ViT-B/32 residual stream activations

	Mean L0 ± s.d.			
Layer	Normal	28x28 patch	56x56 patch	112x112 patch
1	989.19 ± 0.19	989.89 ± 0.31	989.38 ± 0.23	989.27 ± 0.21
2	757.83 ± 0.01	757.79 ± 0.01	757.83 ± 0.01	757.83 ± 0.01
3	1007.89 ± 0.03	1007.88 ± 0.01	1007.88 ± 0.02	1007.89 ± 0.03
4	935.06 ± 1.17	941.41 ± 1.47	938.24 ± 1.44	937.05 ± 1.40
5	965.15 ± 0.28	973.05 ± 1.85	969.00 ± 1.13	966.88 ± 0.78
6	966.36 ± 1.02	1004.33 ± 6.45	987.49 ± 4.37	974.20 ± 1.96
7	1006.62 ± 0.10	1006.89 ± 0.48	1005.78 ± 0.30	1005.89 ± 0.25
8	984.19 ± 0.24	988.70 ± 0.76	984.80 ± 0.41	984.30 ± 0.28
9	965.12 ± 0.15	965.75 ± 0.13	965.54 ± 0.15	965.34 ± 0.17
10	854.91 ± 0.15	855.21 ± 0.10	855.23 ± 0.11	855.08 ± 0.11
11	1141.99 ± 1.20	1145.59 ± 1.19	1141.99 ± 1.20	1141.99 ± 1.20
12	829.09 ± 0.21	829.62 ± 0.05	829.38 ± 0.10	829.19 ± 0.16

Table 7: Mean layer-wise L0 for SAE trained on Pythia-160m-deduped residual stream activations

	Mean L0 ± s.d.					
Layer	Normal	1-gram	2-gram	6-gram	10-gram	30-gram
1	29.50± 0.19	30.24 ± 0.20	29.75 ± 0.20	29.29 ± 0.20	29.20 ± 0.19	29.09 ± 0.20
2	16.18± 0.04	16.20 ± 0.08	15.93 ± 0.06	15.85 ± 0.06	15.86 ± 0.06	15.90 ± 0.06
3	13.25± 0.05	13.06 ± 0.04	12.73 ± 0.05	12.71 ± 0.05	12.76 ± 0.05	12.88 ± 0.05
4	21.41± 0.10	24.84 ± 0.05	22.60 ± 0.08	21.28 ± 0.09	21.13 ± 0.10	21.06 ± 0.10
5	22.31± 0.10	31.62 ± 0.20	25.04 ± 0.08	21.84 ± 0.08	21.57 ± 0.10	21.55 ± 0.10
6	31.98± 0.15	43.69 ± 0.13	38.51 ± 0.09	33.93 ± 0.07	33.40 ± 0.11	33.04 ± 0.15
7	45.89± 0.23	72.17 ± 0.27	67.39 ± 0.44	52.69 ± 0.27	49.18 ± 0.23	46.37 ± 0.22
8	65.53± 0.14	102.59 ± 0.41	101.24 ± 0.56	80.53 ± 0.33	73.63 ± 0.21	67.85 ± 0.15
9	67.55± 0.24	149.03 ± 1.53	130.36 ± 1.18	91.48 ± 0.86	80.23 ± 0.62	71.42 ± 0.38
10	49.12± 0.24	100.54 ± 0.69	92.59 ± 0.65	71.70 ± 0.60	62.64 ± 0.51	53.06 ± 0.32
11	31.57± 0.11	85.71 ± 0.81	75.50 ± 0.79	49.56 ± 0.54	41.42 ± 0.38	34.16 ± 0.20
12	11.03± 0.17	14.94 ± 0.43	13.54 ± 0.40	10.90 ± 0.28	10.42 ± 0.24	10.36 ± 0.21

Interestingly, using just the L0 from layer 6 of the ViT for each image, we can even predict if the semantic image labels from CLIP will misclassify the image, compared to the ground-truth label from ImageNet. We train a simple linear classifier on L0 counts of SAE latents extracted from 9000 images, and verify the results across 10 independent cross validation folds. We also attempt to use the pattern of concept activations, extracted by a partial least squares (PLS) model, to predict the same misclassification. While still robustly above chance, the results for the PLS model are slightly inferior to the raw L0 results. See Table 8 for the mean AUC scores +/- 1 s.d. on this task. We also test both natural and shuffled images, finding that using the L0 as an input we see similar prediction performance for both types of image. Though the actual rate of hallucination (misclassification in this case) is higher in the shuffled images, it is not easier to predict whether a misclassification will occur for a particular sample.

Table 8: Prediction of semantic mislabeling by CLIP (a proxy for hallucination), using a linear classifier trained only on layer 6 ViT L0 scores for each image as input. Natural and shuffled images tested separately. We report mean AUC +/- 1 s.d. over 10 cross validation folds.

	Natural Images	**56x56 Patch Shuffled Images**
L0 Misclassification Prediction	64.8 ± 1.1	61.3 ± 1.6
PLS-Concept Misclassification Prediction	64.5 ± 1.5	61.5 ± 1.3

E Hallucination task specifications and additional results

We use the popular Vectara hallucination leaderboard [24] evaluation methodology to connect patterns of activated concepts in inputs to hallucination in model-generated output. For this task, we test Gemma-2B-IT by inputting the following prompt:

```
You are a chat bot answering questions using data.
You must stick to the answers provided solely by the
text in the passage provided. You are asked the question
"Provide a concise summary of the following passage, covering
the core pieces of information described." <PASSAGE>\n
```

Where `<PASSAGE>` is replaced with a full length source article or story from the Vectara leaderboard dataset (1,006 examples). This prompt is identical to the prompt used when evaluating models throughout the Vectara leaderboard, and we also set the temperature parameter to 0 during generation to match the task specifications. We use the pre-trained HHEM-2.1 model [3] to give a hallucination score to each model-generated output summary. HHEM-2.1 takes provided a source-summary pair and returns a continuous hallucination score from 0.0 to 1.0, with 0.0 indicating minimum hallucination, and 1.0 indicating maximum hallucination. The motivation for this evaluation of hallucination is straightforward: if HHEM-2.1 detects fabricated content in the summary that is not present the source text, then the summary is said to be hallucinated and is given a hallucination score near 1.0. Note that this task does not measure how factually correct the summary is, rather it only measures the faithfulness of the summary to the source. Therefore this task is well-suited to measuring notions of hallucination and alignment with the task objectives.

With these source-summary pairs in hand, we next fit a PLS regression (see Preliminaries section for details on PLS) on the maximum concept activations across all tokens in the source as given by pre-trained Gemma-2B SAEs [5]. We fit a new PLS model independently for layers 1, 7, 11, 13, and 18 on these concept activations, with the continuous hallucination score for the corresponding model-generated summary as the supervised target. We perform 10-fold cross validation to test the PLS model's generalization on new, unseen example concept activations and hallucination scores. To identify key concept activations in the source that drive hallucinations in the model-generated output (according to the PLS model), we compute variable importance in projection (VIP) scores for each concept. VIP is a metric that pools the contribution of a variable across all PLS components, and is calculated for a single concept j as follows:

$$\mathrm{VIP}_j = \sqrt{\frac{d_{\mathrm{SAE}} \sum_{c=1}^{N_{\mathrm{comp}}} \mathrm{SS}_c w_{j,c}^2}{\sum_{c=1}^{N_{\mathrm{comp}}} \mathrm{SS}_c}},$$

where d_{SAE} is the number of SAE concepts (in the case of the Gemma-2B SAEs this is 16,384), N_{comp} is the number of PLS components (4 in this case), $w_{j,c}$ is the weight matrix for projection of the jth SAE concept into the latent space of the cth PLS component, and SS_c is the sum of squares for the portion of the hallucination score explained by the cth PLS component. Specifically, $\mathrm{SS}_c = q_c^2(\mathbf{X}_{\max} w_c)^2$ where q_c are the hallucination score loadings for the cth PLS component.

To test the practical utility of the most important concepts for hallucination prediction in the PLS, we get the 10 concepts with the highest VIP scores and set their concept activations to 0 during Gemma 2B-IT's summary generation. More specifically, we extract the residual stream activations from Gemma 2B-IT at layer 11, run them through the SAE encoder, and set the resulting SAE concept activations for the 10 VIP-identified concepts to 0. We then run these modified concept activations through the SAE decoder to get reconstructed residual stream activations, and sum these modified reconstructions with the reconstruction error vector for the reconstruction of the unmodified concept activations.

Table 13 lists a few examples of corrected hallucinations post-suppression of the top 10 VIP features in the residual stream of layer 11 of Gemma 2B-IT.

We also provide a comparison of our PLS approach for detecting and classifying hallucinations with existing hallucination detection methods. Tables 9, 10, 11, and 12 show the a comparison of the results on several different hallucination benchmarks: TruthfulQA (generation) [36], TriviaQA (generation) [31], CoQA (generation) [55], and TruthfulQA (multiple choice). We run each hallucination benchmark on Gemma-2B-IT using "Language Model Evaluation Harness" framework to provide a fair point of comparison. We evaluted on the same 10 cross validation folds. We report the mean AUC +/- 1 s.d. across folds in Tables 9-12. For each benchmark dataset, the hallucination detection method (rows of the Tables) must predict whether new, unseen examples are hallucinated. The "ground truth" hallucination label for each sample is determined according to the metric(s) associated with each dataset (columns of the Tables). For instance, TruthfulQA (generation) evaluates faithfulness with BLEU [48], ROUGE1 [35], ROUGE2, and ROUGEL scores, TriviaQA evaluates correctness with an exact matching string, CoQA (generation) also uses an exact match score and a modified f1 score, and TruthfulQA (multiple choice) evaluates hallucination in settings with either a single correct answer or multiple correct answers.

We find that our PLS and concept-based approach significantly outperforms these established methods on all but the CoQA (generation) benchmark dataset. Further, our method only uses evaluates the input into the LLM, while the other methods require at least some generation of the output text itself.

F Impact statement

Our work presents and experimentally tests a hypothesis that transformer model hallucination is connected to an increase in high level concepts inferred by the transformer. We also show that certain types of inputs (in our case noisy or experimentally perturbed inputs) increase the the number of concepts imposed on the inputs by the transformer models. We link the pattern of these high level concepts inferred in the input directly to notions of hallucination in the model-generated output, and show that the hallucination rate of these outputs can be controlled through targeted concept suppression. With this in mind, these hypotheses and techniques could be used to positive (reducing hallucinations, improving output faithfulness) or potentially negative ends (adversarial attacks, misalignment with human values, degraded model performance). Therefore, we expect this work and similar work focused on model alignment and faithfulness to have broad societal impacts in the long run. However, interpretability in transformers and LLMs is still at a relatively nascent stage, and we do not present immediately actionable recipes for negatively controlling or adversarially perturbing transformer models at scale. Thus we do not expect immediate negative societal consequences to stem from this work.

G Code availability

Code for the experiments in this paper is deposited at: https://github.com/dblabs-mcgill-mila/noise-to-narrative

Table 9: Comparison of hallucination detection methods on TruthfulQA (generation). Mean AUC +/- 1 s.d. is reported, over 10 cross validation folds, predicting whether an output is hallucinated or not. * = *our method*

	BLEU	**ROUGE1**	**ROUGE2**	**ROUGEL**
Perplexity [56]	50.0±5.3	47.3±5.1	49.9±1.5	45.7±4.4
Energy Score [38]	47.2±3.0	50.5±3.5	49.9±4.5	52.4±3.6
Lexical Similarity [37]	47.0±2.9	49.9±2.8	53.0±5.7	49.1±3.8
Normalized Entropy [39]	51.1±5.5	49.5±4.1	49.4±1.6	48.6±4.3
EigenScore [10]	53.9±5.4	51.8±4.8	47.0±5.9	51.0±4.9
PLS-Concepts (Layer1)*	59.4±6.3	56.4±5.0	54.4±6.7	56.3±5.6
PLS-Concepts (Layer7)*	59.9±5.0	56.6±3.5	55.3±7.2	54.3±4.6
PLS-Concepts (Layer11)*	**60.0±3.6**	56.3±4.3	**58.3±9.1**	57.3±4.0
PLS-Concepts (Layer13)*	57.6±5.5	**58.2±4.7**	56.6±6.1	**58.9±5.6**
PLS-Concepts (Layer17)*	54.6±3.7	57.0±5.2	52.2±4.6	54.9±6.2

Table 10: Comparison of hallucination detection methods on TriviaQA (generation). Mean AUC +/- 1 s.d. is reported, over 10 cross validation folds, predicting whether an output is hallucinated or not.
* = *our method*

	Exact Match
Perplexity	50.6±2.8
Energy Score	50.0±0.1
Lexical Similarity	68.9±2.9
Normalized Entropy	51.0±1.4
EigenScore	66.3±2.5
PLS-Concepts (Layer1)*	66.3±1.3
PLS-Concepts (Layer7)*	65.6±2.3
PLS-Concepts (Layer11)*	**70.8±2.6**
PLS-Concepts (Layer13)*	70.1±2.4
PLS-Concepts (Layer17)*	59.1±1.9

Table 11: Comparison of hallucination detection methods on CoQA (generation). Mean AUC +/- 1 s.d. is reported, over 10 cross validation folds, predicting whether an output is hallucinated or not.
* = *our method*

	Exact Match	**f1**
Perplexity	50.0±0.0	50.0±0.0
Energy Score	50.4±0.4	50.1±0.4
Lexical Similarity	**66.4±1.6**	**59.0±2.0**
Normalized Entropy	49.9±0.1	50.0±0.1
EigenScore	62.3±1.9	51.0±1.3
PLS-Concepts (Layer1)*	52.9±1.9	52.4±0.9
PLS-Concepts (Layer7)*	52.6±2.0	51.7±1.1
PLS-Concepts (Layer11)*	53.9±2.6	52.1±1.1
PLS-Concepts (Layer13)*	53.9±1.5	51.6±1.3
PLS-Concepts (Layer17)*	49.4±1.9	50.5±1.6

Table 12: Comparison of hallucination detection methods on TruthfulQA (multiple choice). Mean AUC +/- 1 s.d. is reported, over 10 cross validation folds, predicting whether an output is hallucinated or not.
* = *our method*

Metric	**One Correct Answer**	**Multiple Correct Answers**
Perplexity	49.1±4.5	49.4±1.9
Energy Score	50.0±4.2	49.8±0.3
PLS-Concepts (Layer1)*	**61.1±2.5**	62.2±3.7
PLS-Concepts (Layer7)*	58.9±5.5	61.6±3.4
PLS-Concepts (Layer11)*	**61.1±2.6**	63.2±3.5
PLS-Concepts (Layer13)*	60.7±3.6	**63.7±4.1**
PLS-Concepts (Layer17)*	52.4±3.5	55.2±4.9

Table 13: Gemma 2B-IT recovered hallucination examples post layer 11 suppression of top 10 VIP concepts

Summaries		**Fabrications**	**Hallucination Score Change**
No Suppression	With Suppression	(no suppression)	
The review is expected to be completed by the end of 2023 and will be the first time that a commissioner has undertaken such a comprehensive review of the quality of life and care of older people in Wales.	The passage is about the review of the quality of life and care of older people living in residential care in Wales by the Older People's Commissioner for Wales, Sarah Rochira.	• The article does not mention a completion date • The article does not claim that this is the first time such a comprehensive review has been conducted	-0.95
Sure, here's a concise summary of the passage: Wenger's contract runs to the summer, and he has been linked with the England manager's job. Southgate has been appointed as England manager and will lead the team against Scotland and Spain in November and December. Wenger is happy that Southgate has a chance to show his quality and believes he has the necessary experience to make the right decisions. Arsenal's Hector Bellerin is set to sign a new contract, but the club is still interested in Barcelona and Manchester City's midfielders, Aaron Ramsey and Thomas Tuchel.	Sure, here's a concise summary of the passage: Wenger's contract runs to the summer, and he has been linked with the England manager's job. Southgate is in contention for the role, and he has been linked to the England national team. Wenger also spoke about the future of Hector Bellerin and Aaron Ramsey.	• Southgate is serving as the interim manager • The matches are in November, not December • Aaron Ramsey plays for Arsenal, not Manchester City • Thomas Tuchel is a manager, not a midfielder, and is not mentioned at all in the original article	-0.82
Jak Trueman's story is a reminder that even in the face of adversity, it is important to stay positive and never give up hope.	**Summary:** Jak Trueman was a 15-year-old schoolboy who was suffering from a rare form of blood cancer called gamma delta T-cell lymphoma. He documented his life and treatment on social media and attracted more than 28,000 Likes on his Facebook page. Despite his illness, Jak remained positive and touched the hearts of many people in West Lothian and beyond.	• Neglects to mention an illness • Overly vague, misses core facts about the article	-0.74

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction explicitly outline our primary contributions, and these accurately reflect the results presented in the paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We include a Limitations subsection beneath the Conclusion section, outlining the assumptions made and how the results might be expected to scale with larger architectures or alternative interpretability approaches.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: We do not present theoretical results. Our paper is focused on empirical experiments.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We utilize pre-trained models, open model architectures, and publicly available datasets, and where appropriate we provide full specifications for all hyperparameters and training setups for models that we train ourselves (Appendix A and C, Preliminaries). We also provide full and detailed specifications for all experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We have provided the scripts necessary to train and evaluate relevant SAEs, as well as scripts to conduct each experiment. Relevant Python packages are also listed. All code is provided in the Supplementary Materials. All data and pre-trained models are available from public sources.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Full experimental details are provided in the Preliminaries, each of the results sections, and Appendices A, C, and E. We also provide full scripts with the exact setup for each experiment.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We present figures 4a, 4b, and 5a with error bars of one standard deviation. We report the meaning of the error bars in the main text. We undertake cross validation for predicted metrics, and provide further fine-grained results in Appendices C and D.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The details about the resources used to execute the experiments are presented in Appendix A.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and are confident that our work is in compliance.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: We discuss potential societal impacts in the Introduction and Conclusion, and we also include a dedicated Impact statement in Appendix F.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We are not releasing any model or dataset as part of this work that has a high risk for misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cited all the datasets and the packages that were primarily used to produce the experimental code.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: We do not release any new assets for public use as part of this work.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: We did not conduct any crowdsourcing experiments or experiments with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: We did not conduct any crowdsourcing experiments or experiments with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: We do not use any LLMs as part of the core method development.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.