NeuronEye: Query-Guided Visual Concept Activation for Vision-Language Reasoning

Ruiyu Yan[1] **Bowen Chen**[2] **Shaowen Wan**[2] **Lin Zhao**[2*]
[1]New York University [2]New Jersey Institute of Technology
lin.zhao.1@njit.edu

Abstract

Current vision-language models (VLMs) encode visual information in dense hidden states where object identity, spatial layout, and local attributes are implicitly entangled rather than explicitly disentangled, limiting their ability to isolate and modulate the specific visual evidence required by a given language query. Inspired by sparse population coding and top-down modulation in biological vision, we introduce *NeuronEye*, a plug-in framework that constructs a sparse, concept-level neuron vocabulary from intermediate VLM representations and selectively activates query-relevant visual concepts during inference. NeuronEye decomposes vision-token states into an overcomplete sparse basis organized by concept-level clusters, uses the language query to activate relevant clusters and localize the patches where selected concepts are expressed, and injects the focused evidence back into vision tokens. A complementary suppression mechanism attenuates dominant perceptual directions to preserve weaker but relevant cues. All operations run in a single forward pass over a frozen VLM backbone. On Qwen2.5-VL-7B, NeuronEye raises CV-Bench overall accuracy by +3.1 with gains of +9.5 on Distance, and improves BLINK Multi-view by +8.3, with similar trends on LLaVA-1.6-7B. These results suggest that sparse neuron vocabularies can serve not only as post-hoc interpretability tools but also as active interfaces for concept-level visual reasoning.

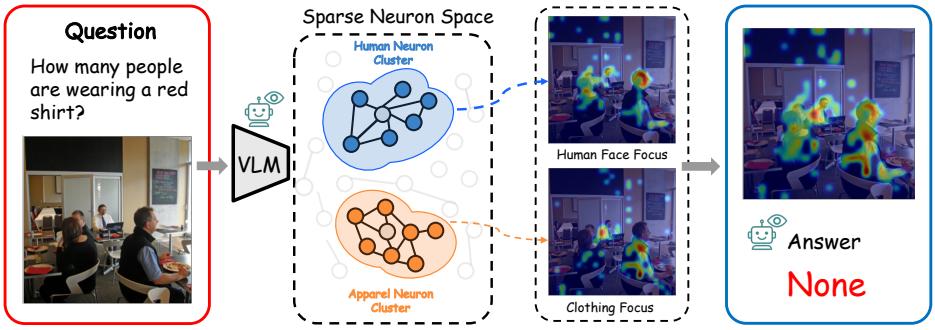


Figure 1: **NeuronEye overview.** Given an image and a language query, NeuronEye decomposes intermediate VLM representations into a sparse neuron vocabulary and identifies query-relevant concept clusters (e.g., *Human* and *Apparel*). Each cluster localizes its corresponding visual evidence in the image, and the focused activations are injected back into the VLM to guide reasoning.

*Corresponding author.
Project page: https://winegorilla.github.io/NeuronEye-Project-Page/

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

1 Introduction

Modern vision-language models (VLMs) align image and text by jointly processing visual and language tokens through learned attention-based multimodal fusion [1, 2, 3]. Although effective on a wide range of tasks, this paradigm encodes visual information in dense hidden states where object identity, spatial layout, local attributes, and background context are not explicitly disentangled. Recent work has shown that VLMs exhibit multi-object reasoning failures remarkably similar to those caused by representational interference in human rapid feedforward vision [4], and that the bottleneck in spatial reasoning often lies in integrating visual information rather than in perceiving it [5, 6]. These findings suggest that dense visual representations may be insufficient for isolating and selectively modulating individual visual concepts according to the question at hand.

The biological visual system addresses this difficulty through two complementary mechanisms. Neurons in primary visual cortex represent natural scenes via sparse population codes, where any stimulus activates only a small subset of neurons while the majority remain silent [7, 8]. At higher cortical levels, this sparsity becomes even more selective: concept cells in the human medial temporal lobe respond to specific persons or landmarks regardless of viewpoint or input modality [9, 10], suggesting that the brain organizes visual information into a sparse, concept-level neural vocabulary whose entries can be independently addressed. Then, top-down attention selectively modulates this vocabulary according to task demands. Treisman's feature integration theory established that separately processed visual features require focused attention to be correctly bound into object percepts [11, 12], and neurophysiological studies have confirmed that prefrontal top-down signals enhance task-relevant neural responses while suppressing competing representations [13, 14]. Together, these mechanisms implement a two-stage strategy: decompose visual input into a sparse concept-level vocabulary, then selectively modulate task-relevant entries to guide downstream perception and reasoning.

Inspired by this perspective, we introduce *NeuronEye*, a framework that constructs a sparse, concept-level neuron vocabulary from intermediate VLM representations and uses language queries to selectively activate relevant visual concepts during inference (Fig. 1). NeuronEye has three core components. *Sparse Neuron Space* (SNS) projects vision-token states into an overcomplete sparse basis via a sparse autoencoder (SAE) and groups the resulting neurons into concept-level clusters, forming an addressable visual concept vocabulary. *Neuron-guided Visual Focus* (NVF) uses the language query, together with image-side evidence, to activate relevant neuron clusters, locate the patches where the selected concepts are expressed, and inject the activated evidence back into the corresponding vision tokens. *Perceptual Concept Suppression* (PCS) complements NVF by attenuating dominant perceptual directions at a later layer, preventing focused activation from suppressing weaker but relevant cues. All operations run in a single forward pass with the VLM backbone and SAE frozen.

We evaluate NeuronEye on four vision-centric benchmarks, demonstrating that it consistently improves structured visual reasoning. Applied to Qwen2.5-VL-7B, NeuronEye raises CV-Bench overall accuracy from 78.5 to 81.6 (+3.1), with gains of +9.5 on Distance and +2.3 on Relation, and improves BLINK Multi-view by +8.3. The method also transfers to LLaVA-1.6-7B with similar trends on spatial and relational sub-tasks. Compared with vision-token reduction and representation-level intervention baselines under the same controlled evaluation protocol, NeuronEye achieves the best overall performance on both CV-Bench and BLINK without finetuning the VLM backbone.

Our contributions are summarized as follows:

- We propose NeuronEye, a concept-level selective modulation framework that decomposes dense visual representations into a sparse neuron vocabulary organized by visual concepts, and activates query-relevant neuron clusters to modulate visual evidence for VLM reasoning.
- NeuronEye operates as a plug-in module over a frozen VLM backbone. After one-time SAE pretraining, only lightweight NVF modules and a PCS scalar are trained, and inference requires a single modified forward pass.
- We show that SAE-derived features can form a structured visual concept vocabulary for query-conditioned selection and localized activation, enabling targeted intervention in VLM reasoning.
- We evaluate NeuronEye on four vision-centric benchmarks across two VLM backbones, demonstrating consistent improvements on spatial and localization-sensitive tasks.

2 Related Works

2.1 Visual Representation Modulation

Prior work has explored visual representation modulation to focus VLM reasoning on relevant visual evidence. Methods such as FastV [15], SparseVLM [16], PruMerge [17], and PyramidDrop [18] remove, merge, or retain patch tokens based on attention or importance scores, determining *where* the model attends. However, each retained token remains a dense unit carrying all visual concepts together; selecting a token retains all its encoded information, including information irrelevant to the query. Representation-level methods such as VISTA [19] adjust *what* is represented by steering dense hidden states, but these modifications are applied uniformly and do not select which visual concept to enhance for a given query. NeuronEye operates at a finer granularity. Rather than selecting which patch tokens to keep or uniformly steering dense hidden states, it decomposes vision-token representations into a sparse concept-level neuron vocabulary and uses the language query to activate relevant concept subsets within localized patches for targeted modulation.

2.2 Sparse Autoencoders in Multimodal Models

Sparse autoencoders (SAEs) have been widely adopted to decompose superposed activations into sparse, interpretable latent features [20, 21]. In this post-hoc setting, SAE features serve as a lens for understanding model internals, revealing human-interpretable directions that correspond to recognizable concepts in the representation space [22, 23]. More recently, several works have moved beyond interpretation toward active intervention, using SAE features to steer model behavior. By amplifying or suppressing selected latent directions, these methods bias model outputs toward desired content or away from undesired behavior [24, 25]. In the multimodal setting, SAVE [26] and SSL [27] apply this steering paradigm to VLMs, using SAE features to identify hallucination-related directions and globally suppress them during generation. While effective for hallucination mitigation, these methods identify a fixed set of SAE features offline and apply them without conditioning on the specific query, and do not organize features into structured groups for selective routing. They therefore do not target query-specific visual reasoning or fine-grained evidence selection. NeuronEye goes beyond steering by organizing SAE-derived visual features into concept-level neuron clusters, selecting relevant clusters conditioned on the language query, and activating them within localized image patches, turning the sparse feature basis into a structured vocabulary for visual reasoning.

3 Method

NeuronEye operates through three stages. Given an image–question pair, *Sparse Neuron Space* (SNS, Section 3.1) first decomposes intermediate visual representations into an overcomplete sparse basis and organizes the resulting neurons into concept-level clusters, forming an addressable visual concept vocabulary. *Neuron-guided Visual Focus* (NVF, Section 3.2) then uses the language query and image-side evidence to activate relevant neuron clusters, localize the patches where selected concepts are expressed, and inject the activated evidence back into the corresponding vision tokens at layer ℓ. Finally, *Perceptual Concept Suppression* (PCS, Section 3.3) attenuates dominant perceptual directions at a later layer ℓ' to preserve representational diversity after focused activation.

3.1 Sparse Neuron Space (SNS)

Sparse Neuron Extraction. Given an image–question pair, we pass the visual input and textual query through the frozen VLM and extract the hidden representation of each image patch token at a designated intermediate layer ℓ, denoted as $\mathbf{x}_{i,p} \in \mathbb{R}^d$, where i indexes the image and p indexes the patch token. We use a SAE to decompose each patch representation into a set of neuron activations (Fig. 2): the SAE projects $\mathbf{x}_{i,p}$ into an overcomplete latent space $\mathbb{R}^D$ with $D = \alpha d$ ($\alpha \gg 1$) using a linear encoder followed by ReLU activation, where each latent dimension corresponds to a *neuron*. A deterministic Top-k operator then retains only the k largest activations and zeros out the rest, producing a sparse code $\mathbf{z}_{i,p}$ with fixed $\ell_0 = k$ sparsity:

$$\mathbf{z}_{i,p} = \mathrm{Top}_k\big(\mathrm{ReLU}(\mathbf{W}_{\mathrm{enc}}\,\mathbf{x}_{i,p})\big). \tag{1}$$

The representation is thus sparse because only k out of D neurons are active for any given patch, and each active neuron carries a scalar activation indicating how strongly that visual concept is

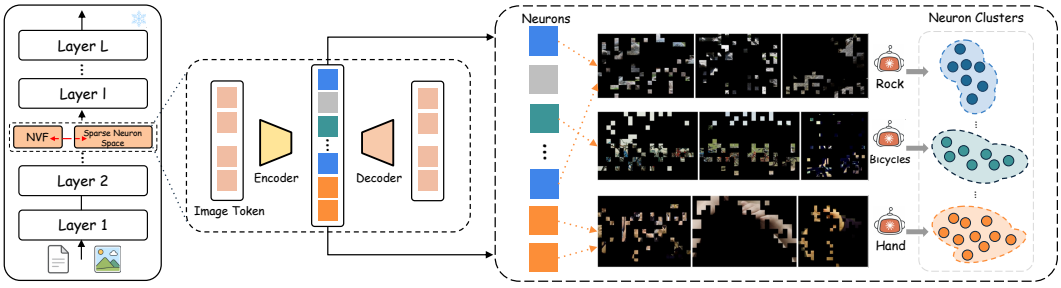


Figure 2: Construction of Sparse Neuron Space. Intermediate vision-token states are encoded by a sparse autoencoder, whose activated neurons are visualized through top-activating patches and grouped into concept-level neuron clusters.

expressed at that patch location. The patch representation is reconstructed as $\hat{\mathbf{x}}_{i,p} = \mathbf{W}_{\text{dec}}\,\mathbf{z}_{i,p}$, where decoder columns are ℓ_2-normalized to prevent scale degeneracy. Training minimizes a patch-level reconstruction objective augmented by an ℓ_1 activation penalty:

$$\mathcal{L}_{\text{SAE}} = \frac{1}{|\mathcal{B}|} \sum_{(i,p)\in\mathcal{B}} \left[\|\mathbf{x}_{i,p} - \hat{\mathbf{x}}_{i,p}\|_2^2 + \lambda_s \|\mathbf{z}_{i,p}\|_1 \right]. \tag{2}$$

where $\mathcal{B}$ denotes a minibatch of image patch tokens. Since the Top-k operator fixes the number of active coordinates, the ℓ_1 term mainly regularizes the magnitude of selected activations.

Neuron Filtering and Neuron Cluster Construction. Not all neurons in the overcomplete latent space are useful. Many are rarely activated across images or respond to visually inconsistent patterns. We therefore filter unreliable neurons and group the remaining ones into neuron clusters. We run the frozen VLM with the trained SAE over the training set and collect patch-level sparse activations $z_{i,p,j}$, where j indexes the neuron. Neurons activated on fewer than $M_{\min}$ distinct images are discarded. For each retained neuron j, we aggregate its activations across patches within each image, select the top-N images where neuron j is most strongly activated, and localize the highest-activating patches by their spatial positions. The resulting patch regions $\mathcal{S}_j$ summarize the visual evidence associated with neuron j. We assign each neuron a short concept label by prompting an external model to summarize the shared visual pattern in $\mathcal{S}_j$ (see Appendix A), then encode labels into text embeddings and apply hierarchical clustering to group neurons into K concept-level clusters $\{\mathcal{C}_k\}_{k=1}^{K}$. These labels are used only for cluster construction; after clustering, each cluster is represented only by its index and the corresponding set of neurons, forming the sparse neuron vocabulary used by NVF.

3.2 Neuron-guided Visual Focus

Given the sparse neuron vocabulary constructed by SNS, NVF uses the language query to activate relevant visual concepts for the current image–question pair. It first predicts which neuron clusters are query-relevant, then localizes the patches where the selected concepts are most strongly expressed, and finally injects the activated evidence back into the corresponding vision tokens.

Query-Guided Neuron Cluster Activation. Each neuron cluster is represented by an index $k \in \{1, \ldots, K\}$ and corresponds to a set of neurons. We obtain a query representation by pooling the textual hidden states and use a lightweight router $g_q(\cdot)$ to produce query-side cluster scores $\boldsymbol{\alpha}^q$ (Fig. 3A). To reduce reliance on language-only priors, a vision-side scorer $g_v(\cdot)$ produces image-grounded cluster scores $\boldsymbol{\alpha}^v$ from visual hidden states:

$$\boldsymbol{\alpha}^q = \sigma(g_q(\text{Pool}(\mathbf{H}_{\text{text}}))) \in [0,1]^K, \qquad \boldsymbol{\alpha}^v = \sigma(g_v(\text{Pool}(\mathbf{H}_{\text{vision}}))) \in [0,1]^K. \tag{3}$$

At inference time, NVF selects the top-K_{sel} clusters according to $\boldsymbol{\alpha}^q$. During training, we obtain supervision labels $\mathbf{y} \in \{0,1\}^K$ by prompting an LLM annotator to identify which neuron clusters are relevant to each text query, based on the cluster descriptions constructed in Section 3.1. The router g_q is then supervised with binary cross-entropy over $\mathbf{y}$, and a per-dimension symmetric divergence term

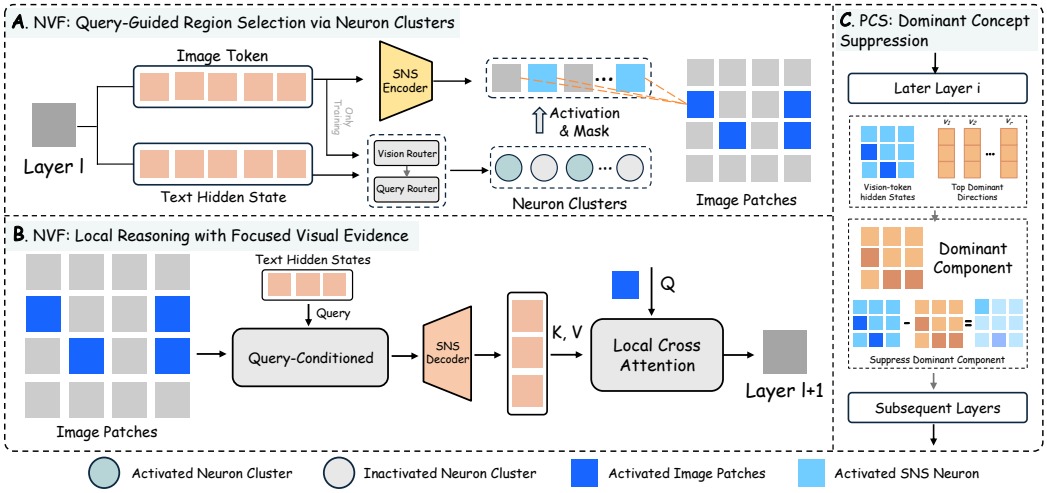


Figure 3: NVF and PCS. NVF selects query-relevant neuron cluster indices and active patches for local cross-attention, while PCS suppresses dominant perceptual components at a later layer.

aligns $\boldsymbol{\alpha}^q$ with $\boldsymbol{\alpha}^v$ to ensure that the selected clusters are grounded in image evidence:

$$\mathcal{L}_{\text{align}} = \frac{1}{2} \sum_{k=1}^{K} \left[\alpha_k^v \log \frac{\alpha_k^v}{\alpha_k^q} + \alpha_k^q \log \frac{\alpha_k^q}{\alpha_k^v} \right]. \tag{4}$$

The final routing objective is $\mathcal{L}_{\text{router}} = \mathcal{L}_{\text{BCE}}(\boldsymbol{\alpha}^q, \mathbf{y}) + \lambda_{\text{align}} \mathcal{L}_{\text{align}}$, which encourages selected cluster indices to be both query-relevant and visually supported.

Local Reasoning with Focused Visual Evidence. For each selected cluster c, let $\mathcal{F}_c$ denote its set of neuron indices. NVF measures how strongly cluster c is expressed at each patch token by summing the activations of its neurons: $a_p^{(c)} = \sum_{j \in \mathcal{F}_c} z_{p,j}$, where $z_{p,j}$ is the activation of neuron j at patch token p. The top-n patches with the highest $a_p^{(c)}$ are selected as active visual evidence for cluster c (Fig. 3B). On these patches, we construct a cluster-specific sparse code $\mathbf{z}_{\text{cluster}}$ by retaining only the neurons in $\mathcal{F}_c$ and masking all others. A query-conditioned refinement module then predicts a gated residual update:

$$\Delta \mathbf{z} = \beta_{\text{ref}} \cdot \mathbf{W}_\Delta \left(\sigma(\mathbf{W}_g \mathbf{z}_{\text{cluster}}) \odot \text{MLP}(\mathbf{q}_{\text{last}}) \right), \tag{5}$$

where $\mathbf{q}_{\text{last}}$ is the hidden state of the last textual token at layer ℓ, and β_{ref} is a learnable scalar controlling the refinement magnitude. The refined sparse code is $\widetilde{\mathbf{z}}_{\text{cluster}} = \mathbf{z}_{\text{cluster}} + \Delta \mathbf{z}$. Each refined code is decoded back to the dense space via the frozen SAE decoder, weighted by its query-side score α_c^q, and summed across selected clusters into $\mathbf{R}_{\text{active}}$.

Finally, NVF injects $\mathbf{R}_{\text{active}}$ into the selected active tokens through localized cross-attention:

$$\mathbf{H}'_{\text{active}} = \mathbf{H}_{\text{active}} + \gamma_{\text{inj}} \cdot \text{LN} \left(\mathbf{W}_o \text{softmax} \left(\frac{\mathbf{Q}_{\text{att}} \mathbf{K}_{\text{att}}^\top}{\sqrt{d}} \right) \mathbf{V}_{\text{att}} \right), \tag{6}$$

where $\mathbf{Q}_{\text{att}} = \mathbf{W}_q \mathbf{H}_{\text{active}}$, $\mathbf{K}_{\text{att}} = \mathbf{W}_k \mathbf{R}_{\text{active}}$, $\mathbf{V}_{\text{att}} = \mathbf{W}_v \mathbf{R}_{\text{active}}$, and γ_{inj} is a learnable injection scale. Only selected tokens are updated; all other vision tokens remain unchanged.

3.3 Perceptual Concept Suppression

PCS addresses a side effect of focused activation. While NVF enhances query-relevant concepts, it may also concentrate vision-token representations along dominant directions, weakening cues such as spatial relations or fine-grained attributes. PCS mitigates this effect at a later layer $\ell' > \ell$ by estimating and attenuating these directions from vision-token geometry (Fig. 3C).

Let $\mathbf{H} \in \mathbb{R}^{N \times d}$ denote the vision-token hidden states at layer ℓ'. We mean-center the tokens as $\bar{\mathbf{H}} = \mathbf{H} - \mathbf{1}\boldsymbol{\mu}^\top$, where $\boldsymbol{\mu} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{H}_i \in \mathbb{R}^d$, to isolate directional structure from the global

mean. We then compute a low-rank decomposition $\bar{\mathbf{H}} \approx \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ and take the top-r right singular vectors $\mathbf{V}_r \in \mathbb{R}^{d\times r}$ as the dominant perceptual directions. The projected dominant component is $\mathbf{P} = \bar{\mathbf{H}}\mathbf{V}_r\mathbf{V}_r^\top$, and PCS suppresses it by:

$$\mathbf{H}' = \mathbf{H} - \eta_{\mathrm{pcs}}\,\mathbf{P}, \tag{7}$$

where $\eta_{\mathrm{pcs}} = \mathrm{softplus}(\eta_{\mathrm{param}})$ is a learnable non-negative scalar initialized near zero, allowing the model to learn the appropriate suppression strength during training. PCS complements NVF: NVF activates query-relevant concepts at layer ℓ, while PCS preserves representational diversity at layer ℓ' by attenuating dominant directions that may suppress weaker visual cues.

4 Experiments

4.1 Dataset

Training data. All training data are drawn exclusively from VQAv2 [28] and are disjoint from the evaluation benchmarks. The SAE for SNS is trained on the full VQAv2 training split. For lightweight NVF modules and the PCS scalar, we construct three independent supervision sets, each with 5K image–question pairs sampled without replacement from the VQAv2 training split and paired with cluster-index labels. These labels are obtained by prompting an LLM annotator to identify query-relevant neuron clusters (Appendix A). For each backbone, we report the mean and standard deviation across the three resulting NeuronEye models.

Evaluation benchmarks. We evaluate on four vision-centric reasoning benchmarks covering spatial understanding, counting, relational perception, localization, and real-world visual reasoning. CV-Bench [5] serves as the primary benchmark, comprising Count, Depth, Distance, and Relation sub-tasks that directly assess structured visual reasoning. BLINK [6] evaluates multi-view reasoning and spatial localization. RealWorldQA [29] and MMStar [30] provide complementary coverage of general visual question answering. No evaluation data are used during NeuronEye training.

4.2 Implementation Details

We use Qwen2.5-VL-7B and LLaVA-1.6-7B as the base models, which are frozen throughout. NeuronEye constructs SNS at layer $\ell = 8$ with $K = 64$ neuron clusters, applies NVF at the same layer, and applies PCS at a later layer $\ell' = 20$. After SNS construction, neuron-cluster assignments are kept fixed, and only the lightweight NVF and PCS scalar modules are updated. The NVF training procedure and hyperparameters are provided (Appendix B). We use greedy decoding for all generation-based evaluations. Experiments are conducted on NVIDIA RTX PRO 6000 GPUs.

4.3 Benchmark Results across VLM Backbones

Table 1 reports results across CV-Bench, BLINK, RealWorldQA, and MMStar. The upper block lists representative VLM baselines for reference; the lower block applies NeuronEye to Qwen2.5-VL-7B and LLaVA-1.6-7B to assess backbone portability. On Qwen2.5-VL-7B, NeuronEye improves CV-Bench overall by +3.1 and BLINK overall by +0.7, with the strongest gains on *Distance* (+9.5), *Multi-view* (+8.3), and *Localization* (+3.3). On LLaVA-1.6-7B, improvements follow a similar pattern on spatial and relational sub-tasks, including *Distance* (+6.9), *Relation* (+12.1), *Multi-view* (+7.7), and *Localization* (+8.1), though performance decreases on *Count* (-8.3) and *Depth* (-3.3). This mixed pattern likely reflects backbone architectural differences in visual tokenization and patch granularity, which affect how patch-level sparse activations map to localized visual evidence. Across both backbones, the gains are consistently strongest on spatial, relational, and localization-sensitive tasks, aligning with the design of NeuronEye as a concept-level selective activation mechanism.

4.4 Comparison with Visual Representation Modulation Methods

Table 2 compares NeuronEye with vision-token reduction and representation-level methods on Qwen2.5-VL-7B under the same evaluation protocol. NeuronEye achieves the best overall performance on both CV-Bench and BLINK, with especially large gains on CV-Bench *Distance* (+9.5) and BLINK *Multi-view* (+8.3). Token reduction methods show consistent degradation on structure-sensitive sub-tasks, likely because discarding or merging patches eliminates visual evidence that

Table 1: Benchmark results across VLM backbones. The upper block lists representative VLMs for reference. The lower block shows NeuronEye applied to LLaVA-1.6-7B and Qwen2.5-VL-7B, with Δ denoting the improvement over each base model.

	CV-Bench					BLINK			Other Benchmarks	
Model	Overall	Count	Depth	Dist.	Rel.	Overall	MV.	Loc.	RealWorldQA	MMStar
Representative VLM backbones										
DeepSeek-VL1 [31]	61.6	59.0	63.2	58.2	68.5	38.1	50.4	37.7	50.5	38.9
Idefics3-8B-Llama3 [32]	67.7	60.5	72.8	67.2	73.4	42.7	45.9	50.8	62.0	49.3
Phi-4 Multimodal [33]	71.7	68.5	74.2	70.8	75.1	49.7	48.1	56.6	61.8	59.7
InternVL3-8B [34]	81.3	70.9	84.8	83.1	89.7	51.3	51.1	58.2	68.2	59.4
Llava-OneVision [3]	76.0	67.3	80.3	78.5	80.8	46.2	57.1	54.9	66.7	68.3
NeuronEye applied to different backbones										
LLaVA-1.6-7B [35]	64.3	63.3	77.8	54.5	63.3	36.6	40.7	38.0	61.6	36.9
+NeuronEye	65.7 $\pm$0.5	55.0 $\pm$0.2	74.5 $\pm$1.0	61.4 $\pm$0.9	75.4 $\pm$0.5	37.2 $\pm$0.7	48.4 $\pm$1.9	46.1 $\pm$0.6	60.7 $\pm$0.7	36.9 $\pm$0.1
Δ	+1.4	-8.3	-3.3	+6.9	+12.1	+0.6	+7.7	+8.1	-0.9	+0.0
Qwen2.5-VL-7B [2]	78.5	67.1	87.2	76.2	87.2	52.1	55.6	53.3	68.5	58.8
+NeuronEye	81.6 $\pm$0.3	67.9 $\pm$0.6	87.0 $\pm$0.4	85.7 $\pm$0.8	89.5 $\pm$0.5	52.8 $\pm$0.2	63.9 $\pm$0.4	56.6 $\pm$0.9	68.8 $\pm$0.4	59.3 $\pm$0.5
Δ	**+3.1**	+0.8	-0.2	**+9.5**	+2.3	+0.7	**+8.3**	**+3.3**	+0.3	+0.5

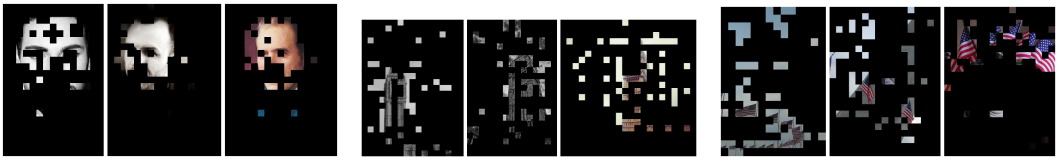
cannot be recovered downstream. Representation-level methods maintain near-baseline performance but offer limited improvement, suggesting that uniform dense-state shifts lack the specificity needed for spatially demanding queries. NeuronEye's gains are concentrated precisely on these structure-sensitive tasks, supporting the hypothesis that concept-level selective activation provides finer control over which visual evidence is enhanced for a given query.

Table 2: Comparison with visual representation modulation methods on Qwen2.5-VL-7B. All methods are evaluated under the same protocol.

	CV-Bench					BLINK			Other Benchmarks	
Model	Overall	Count	Depth	Dist.	Rel.	Overall	MV.	Loc.	RealWorldQA	MMStar
Vision-token reduction methods										
FastV [15]	75.9	64.0	83.5	74.5	85.5	48.4	55.6	49.2	68.8	55.1
SparseVLM [16]	69.8	54.3	74.7	68.8	86.3	46.9	55.6	59.0	50.9	53.8
PruMerge [17]	71.6	55.2	78.0	74.0	83.2	46.8	54.9	51.6	64.3	50.6
MustDrop [36]	73.7	58.6	81.5	74.8	83.9	47.2	55.6	52.5	65.1	53.0
PyramidDrop [18]	72.5	60.2	78.7	70.8	84.2	49.1	55.6	52.5	62.2	50.3
Representation-level intervention methods										
SSL [27]	77.6	66.2	84.5	76.0	87.9	51.3	55.6	52.5	69.5	58.1
SAVE [26]	77.9	66.2	85.7	76.5	87.2	52.3	55.6	54.9	69.0	59.2
VISTA [19]	78.0	66.0	85.8	76.2	88.0	51.7	55.6	53.3	69.4	59.6
NeuronEye	**81.6** $\pm$0.3	**67.9** $\pm$0.6	**87.0** $\pm$0.4	**85.7** $\pm$0.8	**89.5** $\pm$0.5	**52.8** $\pm$0.2	**63.9** $\pm$0.4	56.6 $\pm$0.9	68.8 $\pm$0.4	59.3 $\pm$0.5

4.5 Visualization of Sparse Neurons and Visual Focus

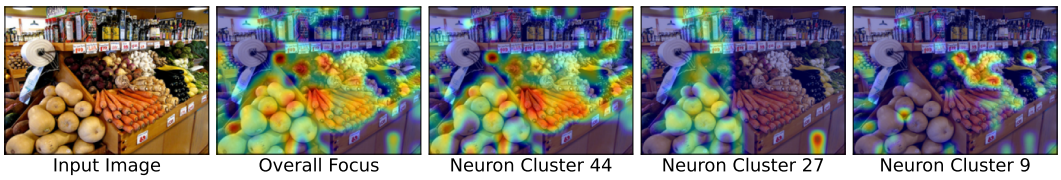
We visualize sparse neuron selectivity and query-guided visual focus to examine how NeuronEye localizes concept-level evidence.



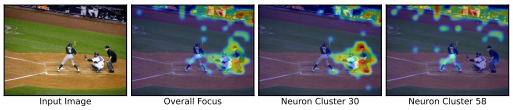
(a) Human-face feature (b) Tower feature (c) Flag feature

Figure 4: Neuron-level feature visualization. Each subfigure shows top-activating patches for one SAE neuron, with non-activating regions masked.

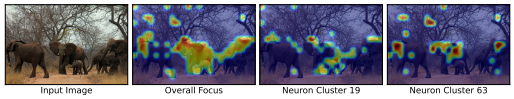
Neuron-level features. Fig. 4 shows that individual SNS neurons activate consistently on semantically coherent regions, such as faces, towers, and flags. These examples support using SNS neurons as a sparse visual vocabulary for concept-level organization. Additional neuron visualizations are provided in the appendix (Appendix C).



(a) Is there organic food in this store?



(b) Are all the players wearing black shirts?



(c) Is the small elephant touching the big elephant?

Figure 5: Question-guided visual focus examples. Each shows the input image, overall focus map, and individual neuron cluster activations.

Fig. 5 visualizes the spatial focus produced by NVF. Given a question, the router selects a small set of neuron clusters and actively triggers the corresponding neurons within these clusters. Their patch-level activations highlight localized image regions expressing the selected sparse features. For the query "Is there organic food in this store?", NVF focuses on shelves and packaged items; for "Are all the players wearing black shirts?", it shifts toward clothing and human-body regions; and for "Is the small elephant touching the big elephant?", it emphasizes the two elephants and their contact region. The resulting focus maps are sparse and localized, indicating that NVF operates through query-selected neuron clusters and patch-level activations rather than global image reweighting. Additional focus examples are provided in the appendix (Appendix D).

4.6 Ablation Studies

Table 3: Ablation study on component contributions.

	CV-Bench					BLINK		
Setting	Overall	Count	Depth	Dist.	Rel.	Overall	MV.	Loc.
Baseline (Qwen2.5-VL-7B)	78.5	67.1	87.2	76.2	87.2	52.1	55.6	53.3
Baseline + FT	76.5 $\pm$0.3	66.6 $\pm$0.2	86.7 $\pm$0.3	69.8 $\pm$0.9	86.5$\pm$0.6	51.1 $\pm$0.1	55.6 $\pm$1.9	55.7 $\pm$0.4
+ Dense Cross-Attn + PCS	79.4 $\pm$0.3	66.2 $\pm$0.6	87.0 $\pm$0.4	79.5 $\pm$1.2	88.8$\pm$0.3	51.3 $\pm$0.3	55.6 $\pm$1.9	54.9 $\pm$1.8
+ Random SNS + NVF	78.6 $\pm$0.4	67.6 $\pm$0.5	86.5 $\pm$0.6	76.6 $\pm$1.0	86.4$\pm$0.6	51.6 $\pm$0.3	55.6 $\pm$1.5	54.9 $\pm$1.0
+ PCS only	78.6 $\pm$0.4	67.3 $\pm$0.4	86.8 $\pm$0.3	76.8 $\pm$1.4	87.2 $\pm$0.6	51.3 $\pm$0.2	55.6 $\pm$1.9	54.9 $\pm$0.4
+ SNS + NVF	80.9 $\pm$0.1	66.8 $\pm$0.6	87.2 $\pm$0.5	84.0 $\pm$0.4	89.5 $\pm$0.5	53.0 $\pm$0.3	55.6 $\pm$2.2	55.7 $\pm$3.0
+ NeuronEye	81.6 $\pm$0.3	67.9 $\pm$0.6	87.0 $\pm$0.4	85.7 $\pm$0.8	89.5 $\pm$0.5	52.8 $\pm$0.2	63.9 $\pm$0.4	56.6 $\pm$0.9

Table 3 summarizes the contribution of each component. Direct fine-tuning of the lightweight modules without sparse intervention (Baseline + FT) does not improve performance and substantially reduces CV-Bench Distance, suggesting that limited VQA fine-tuning alone is insufficient to produce the observed spatial reasoning gains.

Replacing semantically organized neuron clusters with random assignments (Random SNS + NVF) yields near-baseline performance, indicating that the gains do not come merely from added parameters or reconstructed-feature injection. Rather, structured clusters are needed to route sparse visual evidence meaningfully. PCS alone produces only marginal changes, suggesting that attenuating dominant directions is insufficient without query-guided feature selection. SNS + NVF improves CV-Bench Overall and Distance, while the full NeuronEye configuration further improves Distance and BLINK Multi-view. Together, these results support the complementary design of NeuronEye: NVF activates localized query-relevant sparse features, while PCS later attenuates dominant directions to preserve representational diversity and weaker cues.

Table 4 demonstrates the effect of the NVF insertion layer. Performance peaks at layer 8, especially on CV-Bench Distance and BLINK Multi-view. Earlier insertion at layer 4 remains competitive but is slightly weaker, suggesting that the representation may not yet provide sufficient semantic abstraction for reliable cluster routing. Deeper layers from 12 to 20 degrade toward the baseline, likely because spatial details have become increasingly compressed.

Table 4: Effect of the NVF insertion layer. Layer 8 achieves the best results, suggesting a balance between low-level perceptual features and high-level semantic abstraction.

	CV-Bench					BLINK		
Layer ℓ	Overall	Count	Depth	Dist.	Rel.	Overall	MV.	Loc.
4	80.5 ±0.1	67.3 ±0.2	87.0 ±0.6	83.0 ±0.4	88.5 ±0.3	51.6 ±0.3	57.2 ±1.2	56.6 ±1.1
8	81.6 ±0.3	67.9 ±0.6	87.0 ±0.4	85.7 ±0.8	89.5 ±0.5	52.8 ±0.2	63.9 ±0.4	56.6 ±0.9
12	78.3 ±0.4	65.2 ±0.8	85.7 ±0.8	79.5 ±1.2	86.3 ±0.3	51.3 ±0.2	55.6 ±0.1	54.1 ±0.1
16	79.0 ±0.3	67.7 ±0.6	87.2 ±0.2	76.9 ±0.5	87.7 ±0.6	51.8±0.5	56.9±1.9	56.3±0.5
20	78.6 ±0.4	66.8 ±0.3	86.8 ±0.4	77.0 ±1.1	86.9 ±0.6	51.7 ±0.1	55.6 ±1.9	55.7 ±1.3

We further examine the effect of the PCS insertion layer on Qwen2.5-VL-7B while fixing SNS/NVF at layer $\ell = 8$. Table 5 reports results for $\ell' \in \{12, 16, 20, 24\}$ with all other settings unchanged.

Table 5: Sensitivity to the PCS insertion layer on Qwen2.5-VL-7B, with SNS/NVF fixed at layer 8. Results are reported as mean ± standard deviation across three runs.

	CV-Bench					BLINK		
PCS layer ℓ'	Overall	Count	Depth	Dist.	Rel.	Overall	MV.	Loc.
12	80.4 ±0.3	66.0 ±0.6	87.1 ±0.2	82.5 ±1.2	89.8 ±0.7	52.3 ±0.2	62.1 ±1.8	56.0 ±1.7
16	81.0 ±0.3	68.0 ±0.6	87.2 ±0.2	82.8 ±0.9	89.5 ±0.6	52.4 ±0.4	64.0 ±0.9	57.7 ±0.8
20	81.6 ±0.3	67.9 ±0.6	87.0 ±0.4	85.7 ±0.8	89.5 ±0.5	52.8 ±0.2	63.9 ±0.4	56.6 ±0.9
24	81.1 ±0.3	68.0 ±0.8	87.3 ±0.1	82.7 ±0.8	89.6 ±0.6	52.7 ±0.3	63.8 ±1.4	58.0 ±0.8

Layer 20 achieves the highest CV-Bench overall, Distance, and BLINK overall accuracy among the tested positions. Layer 16 performs slightly better on Multi-view, while layer 24 achieves the highest Localization accuracy. Overall accuracy remains similar across layers 16–24, although Distance is more sensitive to placement. We use layer 20 as the default PCS insertion layer.

Table 6: Effect of the number of neuron clusters K. Too few clusters conflate functionally distinct neurons, while too many fragment coherent functional groups and complicate routing.

	CV-Bench					BLINK		
Clusters	Overall	Count	Depth	Dist.	Rel.	Overall	MV.	Loc.
32	80.4 ±0.2	68.4 ±0.5	86.1 ±0.4	82.1 ±1.2	88.2 ±0.5	52.0 ±0.2	57.2 ±1.5	53.9 ±2.6
64	81.6 ±0.3	67.9 ±0.6	87.0 ±0.4	85.7 ±0.8	89.5 ±0.5	52.8 ±0.2	63.9 ±0.4	56.6 ±0.9
128	80.3 ±0.1	66.4 ±1.1	87.1 ±0.4	82.8 ±0.7	88.1 ±0.4	51.7 ±0.3	57.8 ±1.7	58.3 ±1.3

Table 6 examines the effect of the number of neuron clusters. $K = 64$ achieves the best overall trade-off, with the strongest results on CV-Bench Distance and BLINK Multi-view. With fewer clusters ($K = 32$), neurons corresponding to different visual concepts may be grouped together, reducing routing specificity. With more clusters ($K = 128$), neurons corresponding to the same or closely related concept may be split across multiple clusters, making cluster prediction less stable and reducing the coherence of concept-level activation.

5 Discussion

Limitations. NeuronEye improves tasks that rely on localized spatial or relational evidence, but its effectiveness depends on the coverage of the learned neuron vocabulary. Since the SAE and neuron clusters are trained on VQAv2 and then frozen, the sparse neuron vocabulary reflects general-domain visual concepts and may lack coverage of domain-specific ones. This limitation is evident on a medical domain VQA dataset (Appendix E), OmniMedVQA-Mini [37], where NeuronEye reduces overall accuracy from 65.30% to 63.10%, with larger drops on disease diagnosis (-2.80), lesion grading (-2.18), and other biological attributes (-5.51). NeuronEye also relies on LLM-generated cluster-index labels for routing supervision at scale. We manually verified approximately 20% of the generated labels, but misalignment between query intent and selected neuron clusters may persist in

the remainder, potentially degrading the model's performance. Additionally, NeuronEye introduces approximately 20% inference latency overhead and 42% peak memory increase relative to the base model, primarily due to frozen SAE encoding and localized cross-attention (Appendix F).

Broader impacts. NeuronEye may improve visual reasoning and make selected visual evidence easier to inspect. However, it may retain hallucinations and biases from the underlying VLM, and its focus maps do not guarantee correct answers. Its performance decline on medical data also highlights the need for domain-specific validation before use in high-stakes settings.

Future directions. Future work could expand the sparse neuron space using larger and more diverse training corpora, or construct domain-specific neuron vocabularies for specialized settings such as medical imaging. Extending sparse routing across multiple layers may enable coordinated modulation across levels of visual abstraction. Learning neuron clusters directly from data could further reduce reliance on offline concept labeling and improve robustness under distribution shift.

6 Conclusion

We presented NeuronEye, a plug-in framework that decomposes dense VLM representations into a sparse, concept-level neuron vocabulary and selectively activates query-relevant visual concepts at inference, without modifying or retraining the VLM backbone. Experiments across four benchmarks and two backbones demonstrate consistent improvements on spatial, relational, and multi-view reasoning. Our results suggest that sparse autoencoder features can serve not only as interpretability tools but as active interfaces for modulating concept-level visual reasoning.

Acknowledgements

The authors gratefully acknowledge support from the NVIDIA Academic Grant Program for providing computational resources used in this work.

References

[1] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

[2] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[3] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-OneVision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

[4] Declan Campbell, Sunayana Rane, Tyler Giallanza, Nicolò De Sabbata, Kia Ghods, Amogh Joshi, Alexander Ku, Steven M Frankland, Thomas L Griffiths, Jonathan D Cohen, et al. Understanding the limits of vision language models through the lens of the binding problem. *Advances in Neural Information Processing Systems*, 37:113436–113460, 2024.

[5] Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, Austin Wang, Rob Fergus, Yann LeCun, and Saining Xie. Cambrian-1: A fully open, vision-centric exploration of multimodal LLMs. In *Advances in Neural Information Processing Systems*, 2024.

[6] Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A. Smith, Wei-Chiu Ma, and Ranjay Krishna. BLINK: Multimodal large language models can see but not perceive. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024.

[7] Bruno A Olshausen and David J Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

[8] William E Vinje and Jack L Gallant. Sparse coding and decorrelation in primary visual cortex during natural vision. *Science*, 287(5456):1273–1276, 2000.

[9] R Quian Quiroga, Leila Reddy, Gabriel Kreiman, Christof Koch, and Itzhak Fried. Invariant visual representation by single neurons in the human brain. *Nature*, 435(7045):1102–1107, 2005.

[10] Rodrigo Quian Quiroga. Concept cells: the building blocks of declarative memory functions. *Nature Reviews Neuroscience*, 13(8):587–597, 2012.

[11] Anne M Treisman and Garry Gelade. A feature-integration theory of attention. *Cognitive psychology*, 12(1):97–136, 1980.

[12] Anne Treisman. Feature binding, attention and object perception. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 353(1373):1295–1306, 1998.

[13] Robert Desimone, John Duncan, et al. Neural mechanisms of selective visual attention. *Annual review of neuroscience*, 18(1):193–222, 1995.

[14] Behrad Noudoost, Mindy H Chang, Nicholas A Steinmetz, and Tirin Moore. Top-down control of visual attention. *Current opinion in neurobiology*, 20(2):183–190, 2010.

[15] Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *Proceedings of the European Conference on Computer Vision*, 2024.

[16] Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis A. Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. Sparsevlm: Visual token sparsification for efficient vision-language model inference. In *Proceedings of the International Conference on Machine Learning*, 2025.

[17] Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

[18] Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, and Dahua Lin. Conical visual concentration for efficient large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14593–14603, 2025.

[19] Zhuowei Li, Haizhou Shi, Yunhe Gao, Di Liu, Zhenting Wang, Yuxiao Chen, Ting Liu, Long Zhao, Hao Wang, and Dimitris N. Metaxas. The hidden life of tokens: Reducing hallucination of large vision-language models via visual information steering. In *Proceedings of the International Conference on Machine Learning*, pages 35799–35819, 2025.

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

[21] Mateusz Pach, Shyamgopal Karthik, Quentin Bouniot, Serge Belongie, and Zeynep Akata. Sparse autoencoders learn monosemantic features in vision-language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[22] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nicholas L. Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E. Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models With dictionary learning. Transformer Circuits Thread, 2023.

[23] Kaichen Zhang, Yifei Shen, Bo Li, and Ziwei Liu. Large multi-modal models can interpret features in large multi-modal models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

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

[25] Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. Steering Llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 15504–15522, 2024.

[26] Sangha Park, Seungryong Yoo, Jisoo Mok, and Sungroh Yoon. SAVE: Sparse autoencoder-driven visual information enhancement for mitigating object hallucination. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2026.

[27] Zhenglin Hua, Jinghan He, Zijun Yao, Tianxu Han, Haiyun Guo, Yuheng Jia, and Junfeng Fang. Steering lvlms via sparse autoencoder for hallucination mitigation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, 2025.

[28] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2425–2433, 2015.

[29] xAI. RealWorldQA: Evaluating real-world spatial understanding. `https://huggingface.co/datasets/xai-org/RealWorldQA`, 2024. Dataset. Accessed: May 4, 2026.

[30] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. Are we on the right way for evaluating large vision-language models? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[31] Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Hao Yang, Yaofeng Sun, Chengqi Deng, Hanwei Xu, Zhenda Xie, and Chong Ruan. DeepSeek-VL: Towards real-world vision-language understanding. *arXiv preprint arXiv:2403.05525*, 2024.

[32] Hugo Laurençon, Andrés Marafioti, Victor Sanh, and Léo Tronchon. Building and better understanding vision-language models: Insights and future directions. *arXiv preprint arXiv:2408.12637*, 2024.

[33] Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, et al. Phi-4-Mini technical report: Compact yet powerful multimodal language models via mixture-of-LoRAs. *arXiv preprint arXiv:2503.01743*, 2025.

[34] Jinguo Zhu et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

[35] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024.

[36] Ting Liu, Liangtao Shi, Richang Hong, Yue Hu, Quanjun Yin, and Linfeng Zhang. Multi-stage vision token dropping: Towards efficient multimodal large language model. *arXiv preprint arXiv:2411.10803*, 2024.

[37] Yutao Hu, Tianbin Li, Quanfeng Lu, Wenqi Shao, Junjun He, Yu Qiao, and Ping Luo. Omnimed-vqa: A new large-scale comprehensive evaluation benchmark for medical lvlm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

A Offline Interpretation Prompts

Figure 6 shows the offline prompts used for sparse-neuron interpretation and cluster-index supervision. We use Claude 3.7 Sonnet, a multimodal model with vision capabilities accessed through the Anthropic API, as an external annotator. The first prompt assigns short concept labels to sparse neurons based on representative image regions that strongly activate each neuron. The second prompt maps training questions to relevant cluster indices using the offline cluster inventory. The resulting concept labels are used to construct neuron clusters, while the cluster-index labels supervise router training. Neither the external annotator nor its textual annotations are required at inference time.

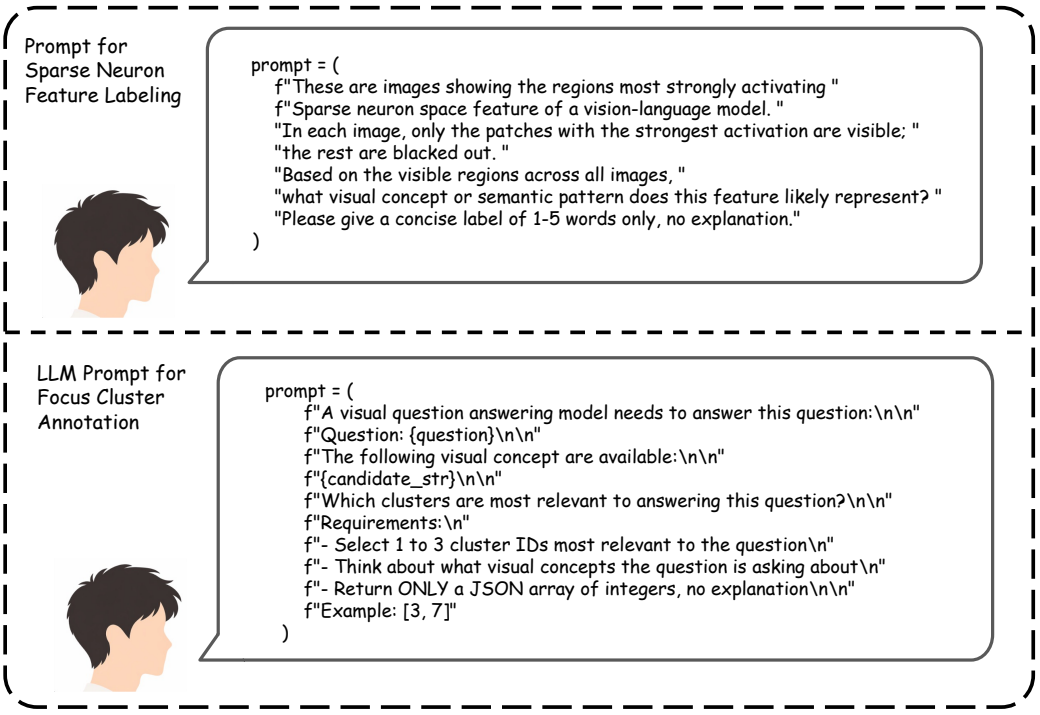


Figure 6: Offline prompts used for sparse neuron interpretation and cluster-index supervision.

B Training, Inference, and Hyperparameter Details

Before training the lightweight modules, we pretrain the SAE on intermediate vision-token activations from the VQAv2 training split and construct the neuron clusters. The SAE and neuron-cluster assignments are then frozen throughout the following two-stage training procedure.

In Stage 1, the router and vision-side scorer are trained using frozen hidden states, optimizing the cluster-index prediction loss and the query–vision alignment loss before any feature injection is introduced. This stage stabilizes the query-to-cluster mapping. In Stage 2, the full NeuronEye pipeline is enabled: NVF performs sparse feature localization, query-conditioned refinement, and localized injection, while PCS suppresses dominant directions at the later layer. The lightweight components are optimized with the combined objective

$$\mathcal{L} = \mathcal{L}_{\text{LM}} + \lambda_{\text{router}} \mathcal{L}_{\text{router}}.$$

Throughout both stages, the VLM backbone, pretrained SAE, and neuron-cluster assignments remain fixed. Only the lightweight NVF modules and the PCS scalar are updated.

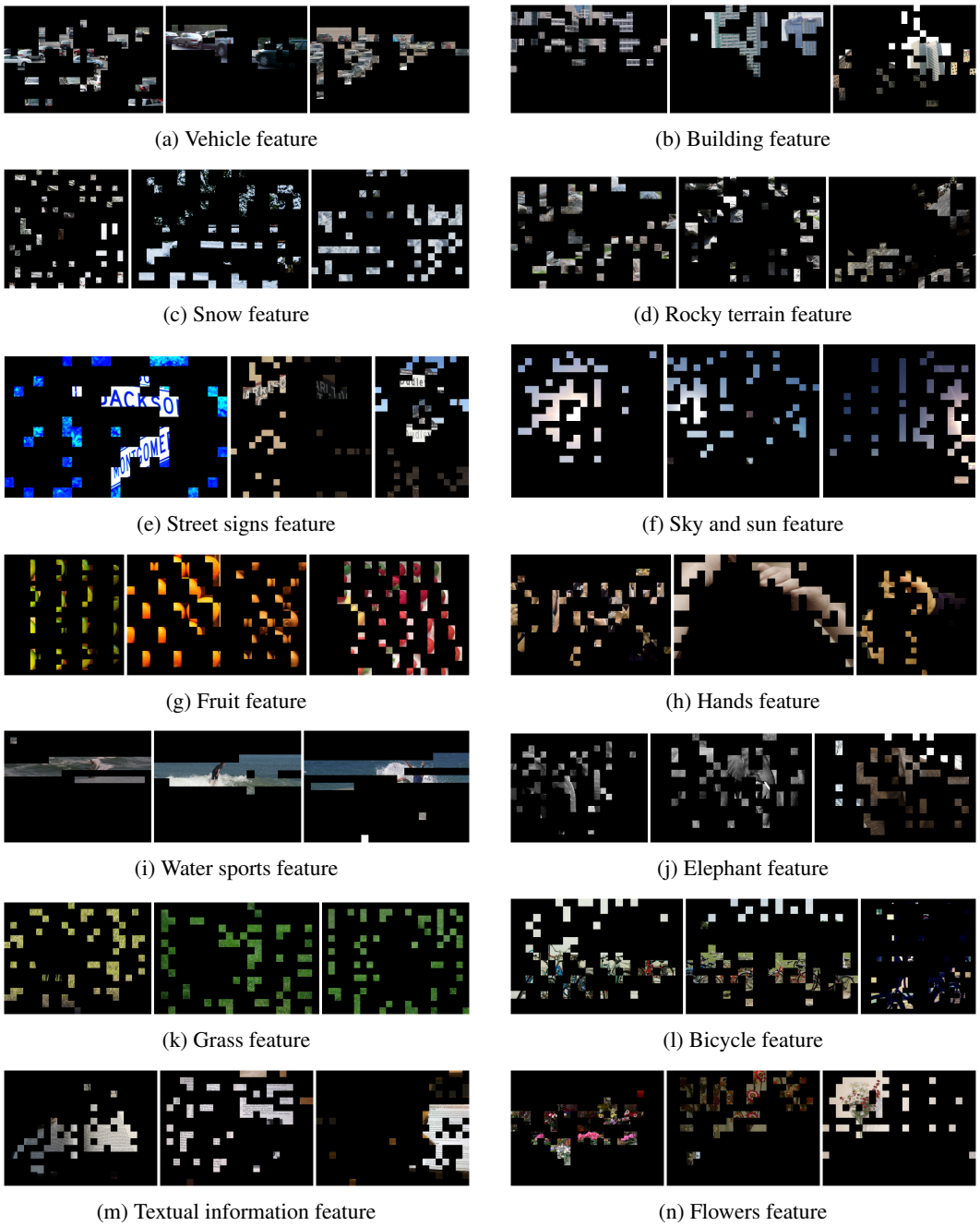
At inference time, NeuronEye runs in a single modified forward pass. NVF selects neuron clusters, localizes and refines the corresponding sparse features, and injects localized updates at layer ℓ, while PCS suppresses dominant directions at layer ℓ'. No external annotator, retrieval module, parameter update, or multi-pass decoding is used during inference.

Table 7: Hyperparameter settings for all experiments.

Component	**Hyperparameter**	**Value**
Sparse Neuron Space	Expansion ratio α	32
	Sparsity level k	32
	Sparsity coefficient λ_s	0.05
	Min. image threshold $M_{\min}$	3
Neuron-guided Visual Focus	Top-n patches per cluster	60
	Top-K_{sel} clusters	5
	γ_{inj} init	0.75
	Bottleneck dim	128
	β_{ref} init	-3.0
PCS	Suppressed directions r	2
	η_{param} init	-3.0
Training	Learning rate	1×10^{-4}
	Gradient accumulation	6
	Training samples	5K
	λ_{router}	0.5
	λ_{align}	0.3

C Additional Sparse Neuron Visualizations

Figures in this section provide additional examples of individual SNS neurons and their top-activating patches. Each visualization shows image regions that strongly activate one sparse latent dimension, with non-activating regions masked. These examples illustrate that many retained neurons correspond to localized and semantically coherent visual patterns, supporting their organization into concept-level neuron clusters.



(a) Vehicle feature

(b) Building feature

(c) Snow feature

(d) Rocky terrain feature

(e) Street signs feature

(f) Sky and sun feature

(g) Fruit feature

(h) Hands feature

(i) Water sports feature

(j) Elephant feature

(k) Grass feature

(l) Bicycle feature

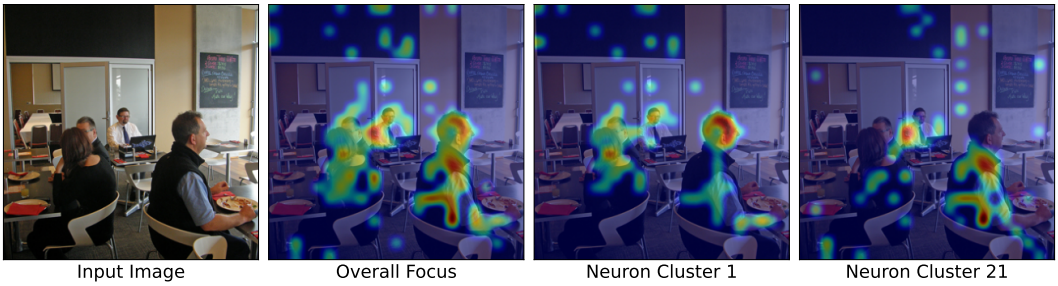
(m) Textual information feature

(n) Flowers feature

Figure 7: Additional sparse neuron visualizations. Each subfigure shows the top-activating patches for a single SNS neuron across multiple images, with non-activating regions masked.

D Additional Question-Guided Focus Examples

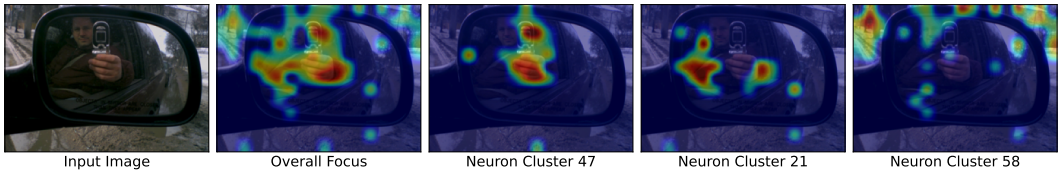
Figure 8 provides additional examples of query-guided visual focus produced by NVF. In each case, NVF selects a small set of query-relevant neuron clusters and increases the activation of neurons within these clusters, yielding cluster-specific activation maps that highlight localized image regions relevant to answering the question.



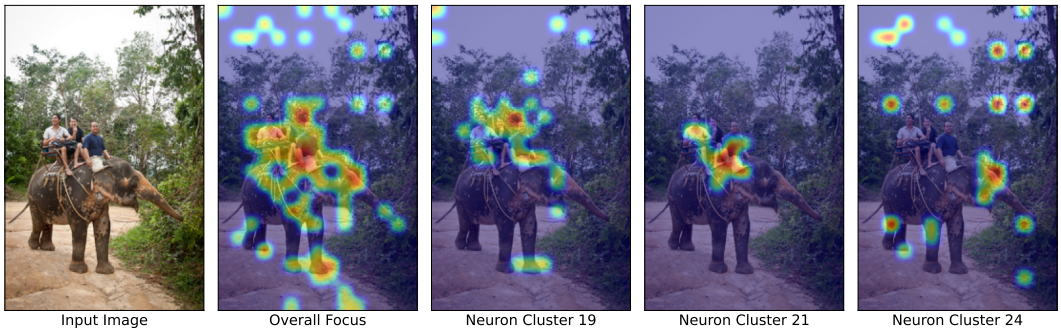
(a) How many people are wearing a red shirt?



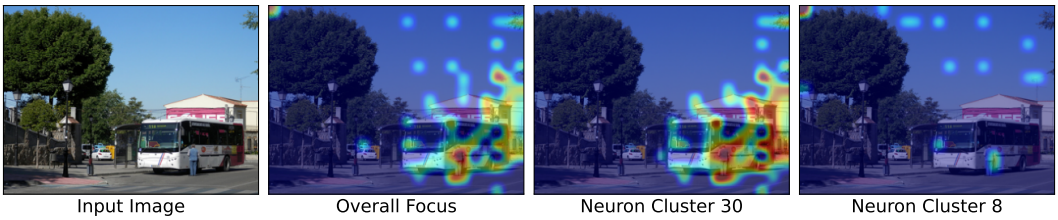
(b) Did the person hit the ball?



(c) Where is the man looking at?



(d) How many people are on the elephant?



(e) Are there passengers waiting to board?

Figure 8: Additional question-guided visual focus examples.

E Additional Domain-Shift Evaluation

Table 8: Performance comparison on OmniMedVQA-Mini by overall accuracy and question type.

Evaluation Setting	**Base Qwen2.5-VL**	**NeuronEye**	**Difference**
Overall Accuracy	65.30%	63.10%	-2.20
Anatomy Identification	48.52%	48.10%	-0.42
Disease Diagnosis	63.74%	60.94%	-2.80
Lesion Grading	57.61%	55.43%	-2.18
Modality Recognition	97.57%	96.18%	-1.39
Other Biological Attributes	71.72%	66.21%	-5.51

Table 8 reports NeuronEye's performance under medical-domain shift on OmniMedVQA-Mini. Overall accuracy decreases from 65.30% to 63.10%, with larger drops in disease diagnosis, lesion grading, and other biological attributes. These results suggest that the sparse neuron vocabulary constructed from general-domain VQAv2 activations may not adequately cover specialized medical concepts, motivating domain-specific SNS construction.

F Efficiency and Overhead

Table 9: Inference overhead relative to the base Qwen2.5-VL model.

Configuration	Latency (s)	Lat. Δ (%)	Peak Mem. (MB)	Mem. Δ (%)
Base (Qwen2.5-VL)	0.382	–	16,857	–
+ SNS + NVF	0.456	+19.4	23,933	+42.0
+ SNS + NVF + PCS	0.457	+19.7	23,933	+42.0

Table 9 summarizes the inference overhead relative to the base model. NeuronEye introduces a moderate latency increase ($\approx$20%), mainly due to frozen SAE encoding and localized cross-attention. The memory increase mainly comes from intermediate activations, routing buffers, and the frozen SAE, rather than additional trainable components. PCS adds negligible extra latency and no additional peak memory in this setting.

Table 10: Parameter breakdown of the added components. Percentages are relative to the 7.6B-parameter base model. Stage 1 and Stage 2 refer to lightweight module training after SAE pretraining and neuron-cluster construction.

Module	**Component**	**Params**	**% of Base**	**Stage 1**	**Stage 2**
SNS	SAE	822.08M	10.79%	Frozen	Frozen
	Neuron Clusters	–	–	Fixed	Fixed
NVF	Router	6.54M	0.09%	Train	Train
	Vision-Side Scorer	6.54M	0.09%	Train	Train
	Projection Layers	12.85M	0.17%	–	Train
	Localized Cross-Attention	51.39M	0.67%	–	Train
	Refinement Module	29.84M	0.39%	–	Train
PCS	η_{param}	1	$\approx$0%	–	Train
Trainable modules		**107.16M**	**1.41%**	–	–

Table 10 reports the parameter breakdown of the added components. The 1.41% parameter overhead refers only to trainable modules. The SAE is trained once for constructing SNS and then kept frozen during lightweight module training and inference. Thus, the frozen SAE contributes non-trainable parameters and memory overhead, while the trainable adaptation cost remains limited to the router, vision-side scorer, projection layers, localized cross-attention, refinement module, and PCS scalar.

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 describe NeuronEye as an inference-time reasoning framework that operates through sparse neuron representations for visual understanding. The claims are supported by the reported experiments, ablations, and limitations, and the scope is restricted to vision-language reasoning tasks evaluated in this work.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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: The paper includes a dedicated limitations discussion. It states that the method depends on the coverage of the learned SNS neuron vocabulary, the quality of neuron clustering and routing supervision, and the general-domain nature of the VQAv2-trained sparse vocabulary. It also reports domain-shift degradation and inference/memory overhead.

   Guidelines:

   - The answer [N/A] 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: [N/A]

   Justification: The paper does not present formal theoretical theorems or proof-based claims. The contribution is methodological and empirical.

   Guidelines:

   - The answer [N/A] 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: The paper describes the base model, SNS construction, training data, evaluation benchmarks, ablation settings, decoding strategy, and implementation details necessary to reproduce the main results.

   Guidelines:

   - The answer [N/A] 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: The experiments use publicly available datasets and open-source base models. The implementation, configuration files, data preparation instructions, and evaluation scripts are publicly available through the project page: `https://winegorilla.github.io/NeuronEye-Project-Page/`. These resources support reproduction of the main experimental results.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental section reports the base VLM, plug-in layer, sparse latent configuration, neuron clustering setting, fine-tuning subset, trainable modules, evaluation datasets, decoding setup, and ablation variants. Additional implementation details are provided in the appendix.

   Guidelines:

   - The answer [N/A] 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: The paper reports mean and standard deviation across repeated runs in which the lightweight NeuronEye modules are trained on different randomly sampled 5K fine-tuning subsets. We specify this source of variation and use the same evaluation protocol, benchmark splits, and decoding strategy for the base model, NeuronEye, and all ablation variants.

Guidelines:

- The answer [N/A] 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 paper reports the GPU setup used for training and evaluation, along with latency, memory overhead, and parameter footprint comparisons between the base model and NeuronEye variants.

   Guidelines:

   - The answer [N/A] 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: The work uses publicly available datasets and models for visual reasoning evaluation and does not involve private data collection, human-subject experiments, or deceptive deployment. The experiments are conducted for research purposes and follow standard responsible ML practice.

Guidelines:

- The answer [N/A] 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: The paper discusses that sparse neuron reasoning may improve interpretability and reliability of vision-language models, while also noting that stronger visual reasoning systems may inherit risks from the underlying VLMs, including hallucination, bias, and misuse in high-stakes settings.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release a new pre-trained foundation model, image generator, or scraped dataset with high misuse risk. The method builds lightweight reasoning modules on top of existing VLMs and uses standard public benchmarks.

    Guidelines:

    - The answer [N/A] 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: The paper properly credits the datasets, benchmarks, and model checkpoints used in the experiments through citations to their original papers or public release pages. All assets are publicly available resources and are used only for training or evaluation. The released code will provide instructions for obtaining each asset from its official source and will preserve the corresponding license and terms-of-use information.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: The paper does not introduce a new dataset or benchmark. The learned modules and implementation are methodological artifacts rather than a new standalone dataset asset.

    Guidelines:

    - The answer [N/A] 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: [N/A]

Justification: The paper does not involve crowdsourcing experiments or human-subject data collection.

Guidelines:

- The answer [N/A] 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: [N/A]

Justification: The paper does not involve human-subject research, participant recruitment, or private human data collection.

Guidelines:

- The answer [N/A] 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 rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: The paper uses a vision-language model as the base model and, where applicable, uses a VLM-based annotator to assign semantic descriptions to top-activating SNS features. These uses are part of the methodological pipeline and are described in the method and implementation sections.

Guidelines:

- The answer [N/A] 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 in the NeurIPS handbook for what should or should not be described.