# SUPER NEURONS ANSWER BEFORE VLMS SPEAK

**Anonymous authors**
Paper under double-blind review

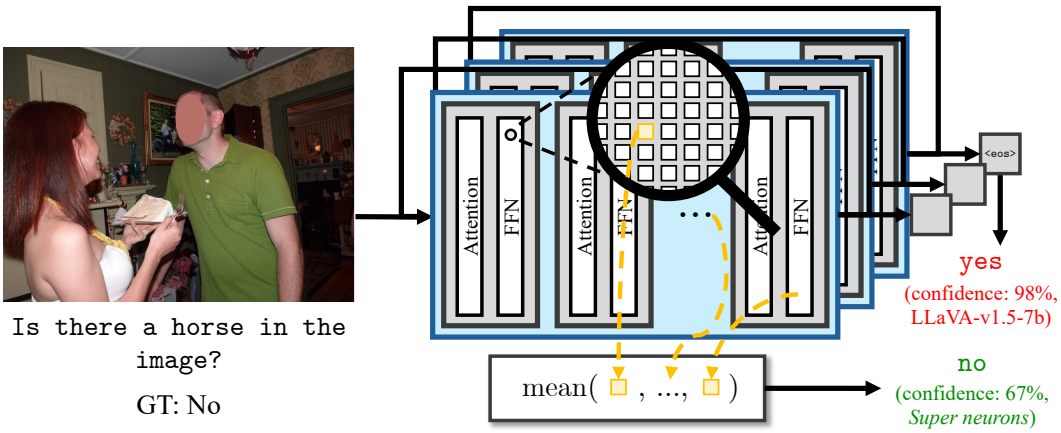



Figure 1: We find *super neurons* inside the LLMs of VLMs that answer better and faster than the full base models themselves on categorical visual question answering benchmarks. A handful of units (*e.g.* 10) are sufficient to achieve such performance. We extract *super neurons* from the hidden states of the LLM, during prefill encoding, bypassing the autoregressive generation of the LLM and dramatically improving runtime. This example is from LLaVA-v1.5-7b's *super neurons* on POPE.

## ABSTRACT

Vector-level representations in pretrained vision-language models can be repurposed for prediction, but how small can these representations be? We show that direct scalar readouts from a handful of neural units (*e.g.* 10) in the LLMs of VLMs can outperform the full, billion-parameter model from which they are selected on categorical visual question-answering tasks. Notably, these *super neurons* can be identified in a gradient-free setting, using a single GPU. We thoroughly analyze *super neurons* and find that they satisfy key properties of effective predictors: they do not seem to exploit spurious biases and generalize well. Therefore, they enable diverse applications. Since they are discovered during prefill encoding, their predictions are able to bypass the autoregressive generation process of the LLM, dramatically reducing runtime by up to 27.8× on Qwen3-VL-8b-Instruct. Since they are sparse, they enable lightweight finetuning, preserving some of the VLM's general capabilities while improving its performance on a target task. Our code will be released upon acceptance.

## 1 INTRODUCTION

Vision-language models (VLMs) have proven successful on diverse benchmarks requiring image understanding and textual generation (Liu et al., 2023; 2024; Qwen Team, 2025; Llama Team, 2024; Beyer et al., 2024; Liu et al., 2025; Chen et al., 2024; Zhang et al., 2023; Kim et al., 2024). Interestingly, while their computation is distributed across the billions of parameters that compose them, it is still possible to identify internal components carrying representations able to further improve their performance (Mitra et al., 2025; Kang et al., 2025b; Skean et al., 2025; Yu and Lee, 2025). For example, a few of their attention heads are able to accurately localize objects (Kang et al., 2025b).

Based on this evidence, we investigate how small and local these components can be. Our answer is that a handful of scalars is sufficient for categorical visual question answering (CVQA). Specif-



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

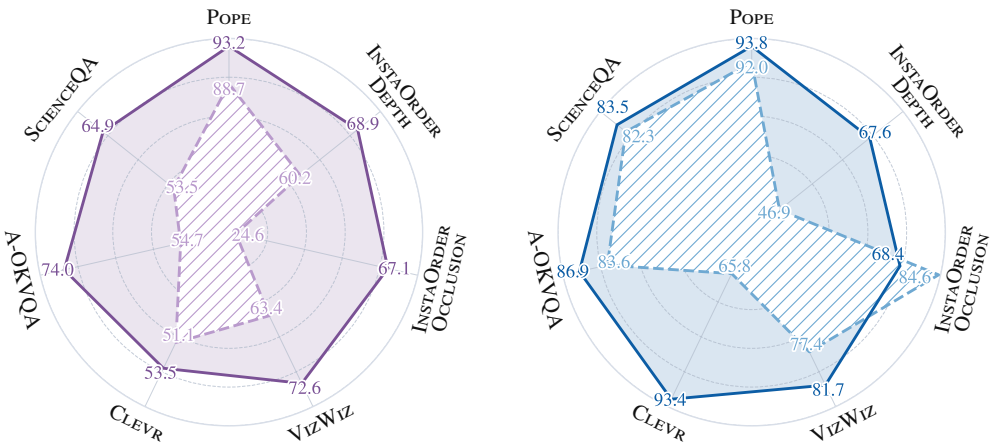



Figure 2: Accuracy comparison between 10 *super neurons* (solid lines) and their base models: on LLaVA-v1.5-7b and Qwen3-VL-8b-Instruct (purple and blue hashed lines, respectively).

ically, we discover that ensembling the scalar outputs of 10 neural units can independently support prediction and even outperform the full base model from which they are extracted.

Using a small probing set, we rank the output of the neurons in the large language model (LLM) of the VLM by measuring their cross-entropy with respect to the intended answer to a task. This gradient-free approach quickly and precisely identifies *super neurons*, *i.e.*, units that outperform the model on a given benchmark. At inference, a simple readout from a small ensemble of these *super neurons* (SNs) is able to accurately solve CVQA benchmarks (fig. 1). Since our algorithm operates during prefill encoding, *super neurons* benefit from *extreme early exiting* by emitting their answers before the autoregressive generation process of the LLM of the VLM even starts. This translates into substantial runtime reduction.

We thoroughly investigate the properties of these *super neurons*. Although relying on single scalars, we discover that these units display attributes of strong predictors: they do not show signs of exploiting spurious biases, do not seem to overfit, and exhibit generalization capabilities. Surprisingly, *super neurons* even obtain strong performance on spatial understanding benchmarks, where their base VLMs otherwise struggle (fig. 2).

These facts are non-trivial. First, it is noteworthy that highly predictive vision-text units can be found in the LLMs of VLMs whose vision encoders and text decoders were initially trained separately. Second, *super neurons* achieve strong performance in benchmarks where the base VLM shows below-chance performance, or exploits dataset statistics. This suggests that, although the relevant information is locally encoded in *super neurons*, it remains underutilized at scale. This is especially intriguing since we report that *super neurons* have a greater causal influence than other units on the model's output.

Although discovered on CVQA benchmarks, we show that *super neurons* can induce transferable improvements. Specifically, we show that finetuning the units contributing to the 10 best *super neurons*, in a low-data regime, not only improves the model's performance on the target task, but also on related open-ended tasks, while preserving the generalist capabilities of the model. These desirable properties cannot be achieved via LoRA (Hu et al., 2022) or full finetuning in the same setting.

Our main findings are as follows:

- Using CVQA probing data naturally uncovers task-wise expert units in VLMs in a gradient-free fashion. We name these units *super neurons*.
- As few as 10 *super neurons* outperform the vanilla model on a diverse CVQA benchmark suite.
- We thoroughly investigate the capabilities of *super neurons* through extensive benchmarks and ablation experiments.

- Although discovered in a CVQA setting, *super neurons* exhibit desirable properties that expand their impact. For example, they allow extreme early exiting and lightweight finetuning that maintains the generalist behavior of the base VLM.

## 2 Related work

**Improving pretrained representations.** The traditional VLM framework (Liu et al., 2023; Li et al., 2023a; Radford et al., 2021; Zhang et al., 2023) can be modified to incorporate stronger visual encoders (Liu et al., 2025; Beyer et al., 2024; Tschannen et al., 2025; Tang et al., 2025; Tong et al., 2024) or stronger language decoders (Llama Team, 2024; Qwen Team, 2025) to improve model performance. But in both cases, this requires aligning the representations of the modules, which is costly (Lou et al., 2025; Liu et al., 2023). In this regard, it is sometimes easier to post-train a VLM via finetuning to improve its performance on a specific downstream visual task (Pang et al., 2024), or through prompt tuning (Lester et al., 2021). However, this requires expensive computational resources due to the size of the LLM and often ends up specializing the model to a specific target task. This can be amortized by training linear probes on intermediate layers, each exhibiting different processing functions (Yu and Lee, 2025; Skean et al., 2025) or even on a subset of neurons, most suited for the task (Gurnee et al., 2023). Prior work reporting highly efficient units on LLMs first specializes the model using prompt tuning (Wang et al., 2022). Moreover, most notable approaches uncovering neuron semanticity mostly focus on interpretability, not performance improvement (Huben et al., 2024; Templeton et al., 2026; Gurnee et al., 2023).

Nonetheless, all these approaches still attach learnable components to the model's states, impeding direct readout from the units themselves, and potentially obfuscating some of the underlying signal during this decoding. By contrast, we propose a simple gradient-free approach that repurposes hidden states of the LLM of the VLM as categorical predictions, improving the performance of the model, while providing a direct, transparent readout from the neural units themselves without training any additional module.

**Discovering pretrained representations.** To avoid altering the model's parameters and representations, gradient-free approaches have been proposed. While simply better prompting (Jiang et al., 2022) can lead to better performance, visual approaches greatly benefit from leveraging learned representations in the attention mechanism of the LLM. Specifically, some attention maps of the LLM of the VLM show object-centric properties enabling native referring image segmentation (Kang et al., 2025a). Closer to our motivations, task vectors (Hojel et al., 2025) have been shown to exist in VLMs and have been used as a sparse ensemble of high-dimensional categorical classifiers (Mitra et al., 2025). Early deep learning works show that unit-level binary categorical predictions are possible with *unimodal* ConvNets (Le et al., 2012) and LSTMs (Radford et al., 2017), but not at scale, and solely for a single well-defined downstream task. Methods targeting neuron-level representations, like neural tagging (Oikarinen and Weng, 2023) leverage extracted information for explainability, but not performance improvement.

In this work, we show that multimodal transformers encode diverse and highly performant task-level signals in the outputs of individual neurons. Unlike SAVs, which behave at the vector level, we shift the discovery of *super neurons* to the scalar level, offering insights that a few of the VLMs' units locally possess strong pretrained representations.

## 3 Method

In this section, we describe how to identify task-wise expert units that we call *super neurons*. We deliberately keep the algorithm simple to make the relationship between each unit's hidden state and its final prediction as transparent as possible.

**High-level algorithm.** Our method starts by gathering a small probing set of CVQA data on which a forward pass of the VLM is computed and hidden states saved. Each neuron's activation is considered as its binary prediction. The $k$ neurons with the lowest binary cross-entropy (BCE) relative to the expected answer are designated as *super neurons*. Each prediction is determined as the mean

**Algorithm 1** Probing and inference pseudo-code using *super neurons*. `pytorch` style.

```
def probe_sns(hs: Tensor, gt: Tensor, k: int, tau: float):
    mu, sigma = hs.mean(0), hs.std(0)                    # [L, D]
    z = flatten((hs - mu) / sigma, dims=(1, 2))          # [N, LD]
    loss = BCE(sigmoid(stack([z, -z])), gt).mean(1)      # [2, LD]
    signs = where(argmin(loss, 0) == 0, +1, -1)          # [LD]
    queue = argsort(min(loss, 0), ascending=True)        # [LD]
    sns = []
    for c in queue:  # greedy decorrelation
        if all(abs(corr(z[:, c], z[:, s])) < tau for s in sns):
            sns.append(c)
        if len(sns) == k:
            break
    return sns, signs[sns], mu, sigma

def infer_sns(hs: Tensor, sns, signs, mu, sigma) -> Tensor:
    z = flatten((hs - mu) / sigma, dims=(1, 2))          # [N, LD]
    z_bar = (signs * z[:, sns]).mean(1)                  # [N]
    return z_bar > 0                                     # [N]

sns, signs, mu, sigma = probe_sns(hs, gt, 10, 0.7)
sns_val_preds = infer_sns(val_hs, sns, signs, mu, sigma)
```

vote of *super neurons*. We validate their effectiveness on the validation set of each dataset. The full pseudo-code is stated in algorithm 1.

### 3.1 Preliminaries

**Problem formulation.** Given a set of neurons $\mathcal{N}$ of the LLM of a VLM that solves task $\Gamma$, we search for a small subset $\mathcal{S}^\star \subseteq \mathcal{N}$ such that $|\mathcal{S}^\star| \ll |\mathcal{N}|$, and such that $\mathcal{S}^\star$ performs at least as well as the full model. We call this subset *super neurons*.

**Inner workings of a VLM.** Let $H$, $W$ be the height and width of an input image and $t \in \mathbb{N}$ be the length of a tokenized prompt. Let $D$ be an arbitrarily fixed latent dimension and $t_{\text{vis}}, t_{\text{txt}} \in \mathbb{N}$ be the lengths of sequences of image and text tokens, respectively. We define a VLM as a combination of vision and text encoders $E_{\text{vis}} : \mathbb{R}^{H\times W\times 3} \to \mathbb{R}^{t_{\text{vis}}\times D}$ and $E_{\text{txt}} : \mathbb{N}^t \to \mathbb{R}^{t_{\text{txt}}\times D}$. Their outputs are concatenated and fed to an LLM $f_\theta : \mathbb{R}^{(t_{\text{vis}}+t_{\text{txt}})\times D} \to \mathbb{R}^{(t_{\text{vis}}+t_{\text{txt}})\times D}$. Given a grounding image $\mathbf{I} \in \mathbb{R}^{H\times W\times 3}$ and a tokenized prompt $\mathbf{T} \in \mathbb{N}^t$, a full VLM forward pass is defined as follows:

$$\hat{\mathbf{Y}}^{t+1} = f_\theta(\hat{\mathbf{Y}}^t), \quad \text{where } \hat{\mathbf{Y}}^0 = E_{\text{vis}}(\mathbf{I})||E_{\text{txt}}(\mathbf{T}). \tag{1}$$

Here, $\cdot||\cdot$ denotes the concatenation operator. $\hat{\mathbf{Y}}^{t+1}$ can be auto-regressively fed into $f_\theta$. This process halts when the LLM generates an `<eos>` token.

**Hidden states.** We define a hidden state of an $L$-layered LLM, as $\mathbf{H}_l \in \mathbb{R}^D$. They represent the output values of the final feed-forward network added to the residual stream at its $l$-th transformer block. For conciseness, we omit the subscript when referring to all $L$ hidden states, *i.e.*, $\mathbf{H} \in \mathbb{R}^{L\times D}$.

### 3.2 Identifying a set of *Super neurons* on a probing set

Given a task $\Gamma$ to solve, we construct a probing set $\mathcal{P}^\Gamma = \left\{\left(\mathbf{I}_n^\Gamma, \mathbf{T}_n^\Gamma\right), \mathbf{Y}_n^\Gamma\right\}_{n=1}^{N\in\mathbb{N}}$ to uncover *super neurons* in the LLM $f_\theta$ of the VLM. We record the hidden states from the last token of the prefill from all layers of $f_\theta$ for each sample in the probing set:

$$\mathbf{H}_n^\Gamma = f_\theta\left(E_{\text{vis}}\left(\mathbf{I}_n^\Gamma\right)||E_{\text{txt}}\left(\mathbf{T}_n^\Gamma\right)\right) \in \mathbb{R}^{L\times D}, \quad \forall\left(\mathbf{I}_n^\Gamma, \mathbf{T}_n^\Gamma\right) \in \mathcal{P}^\Gamma. \tag{2}$$

For brevity, we omit the sample index $n$ and refer to the hidden states for all samples as $\mathbf{H}^\Gamma \in \mathbb{R}^{N\times L\times D}$. The key of our method lies in converting individual neural representations from $\mathbf{H}^\Gamma$ into interpretable predictions without relying on learnable modules. We first standardize each unit with its own statistics over the probing set:

$$\mathbf{Z}^\Gamma = \frac{\mathbf{H}^\Gamma - \boldsymbol{\mu}}{\boldsymbol{\sigma}} \in \mathbb{R}^{N\times L\times D}, \qquad \boldsymbol{\mu}, \boldsymbol{\sigma} \in \mathbb{R}^{L\times D}, \tag{3}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are the per-unit mean and standard deviation on $\mathcal{P}^{\Gamma}$, kept frozen thereafter.

The crux of our approach is to interpret each normalized activation $z_{l,d}$ as a logit for the positive class and score it using binary cross-entropy against the ground truth. Since there are two possible orientations for each neuron, we choose the sign $s_{l,d}$ which results in the minimal cross-entropy:

$$\ell_{l,d} = \min_{s_{l,d} \in \{+1,-1\}} \frac{1}{N} \sum_{n=1}^{N} \mathrm{BCE}\big(\mathrm{sigmoid}(s_{l,d} \cdot z_{n,l,d}), \mathbf{Y}_n^{\Gamma}\big) \in \mathbb{R}, \tag{4}$$

where $\boldsymbol{\ell} \in \mathbb{R}^{L \times D}$ is the loss of each individual neuron of $f_\theta$ on $\mathcal{P}^{\Gamma}$.

We then select the top-$k$ performing units as *super neurons*. To improve the diversity of the selected neurons, we exclude pairs of units with highly correlated predictions via a greedy algorithm described in algorithm 1:

$$\mathcal{S}^{\star} = \underset{(l,d) \in \mathcal{N}}{\text{argmin-}k}(\boldsymbol{\ell}) \subset \mathcal{N}, \quad \text{such that,} \quad |\mathrm{corr}(z_i, z_j)| < \tau, \ \forall\, i \neq j \in \mathcal{S}^{\star}, \tag{5}$$

where $\tau = 0.7$. Choosing a small $k$ trivially enforces $|\mathcal{S}^{\star}| \ll |\mathcal{N}|$ and proves sufficient in practice.

### 3.3 Inference and evaluation using *Super neurons*

However, simply evaluating selected *super neurons* $\mathcal{S}^{\star}$ on $\mathcal{P}^{\Gamma}$ does not guarantee their generalization capabilities. Therefore, it is essential to evaluate the probed *super neurons* on a validation set. Akin to $\mathcal{P}^{\Gamma}$, we define the validation set as $\mathcal{V}^{\Gamma} = \left\{ \left(\mathbf{I}_n^{\Gamma}, \mathbf{T}_n^{\Gamma}\right), \mathbf{Y}_n^{\Gamma} \right\}_{n=1}^{N \in \mathbb{N}}$.

We record the hidden states $\mathbf{H}^{\Gamma}$ of $f_\theta$ on $\mathcal{V}^{\Gamma}$ obtained during prefill, and standardize them using the statistics from the probing stage. Then, we aggregate the $k$ selected activations considering each standardized, signed activation from *super neurons* as logits for independent binary predictions, which is equivalent to averaging their logits.

## 4 Experiments

In this section, we empirically document the existence of *super neurons* under CVQA probing. Extensive experiments highlight both the capabilities and applications of these units.

### 4.1 Experimental settings

**Datasets.** We validate our approach on seven diverse categorical visual question answering datasets: Pope, for object hallucination (Li et al., 2023b); InstaOrder (Occ.), for instance-wise occlusion understanding (Musacchio et al., 2025); InstaOrder (Depth), for instance-wise depth understanding; VizWiz, for broad visual understanding (Gurari et al., 2018); Clevr, a synthetic dataset for spatial understanding (Johnson et al., 2017); A-OKVQA, a general knowledge multiple choice question (MCQ) dataset (Schwenk et al., 2022); ScienceQa, an MCQ dataset for mathematics and scientific reasoning (Lu et al., 2022). We provide details for each dataset, including their prompt templates in appendix A.1.

**Probing sets.** We construct probing sets of size $N = 3{,}000$ for each of the aforementioned datasets. We randomly sample them from their respective training sets. By directly using samples from training sets, we ensure *no overlap* between probing and validation data. We make a single exception for VizWiz, which contains fewer than 1K categorical samples in its train set. We convert A-OKVQA and ScienceQa into a series of binary questions allowing our method to be applied indistinguishably as for other datasets. We balance the probing set of each dataset to ensure that each class is evenly represented and to avoid biasing the selected *super neurons*.

**Evaluation.** We evaluate base models and *super neurons* on the official validation sets using traditional accuracy, precision, recall and F1, treating "yes" as the positive class. We report accuracy and F1 in the main manuscript, detailing the full benchmarks in the supplementary materials. The definitions of these metrics are provided in appendix section C.1.

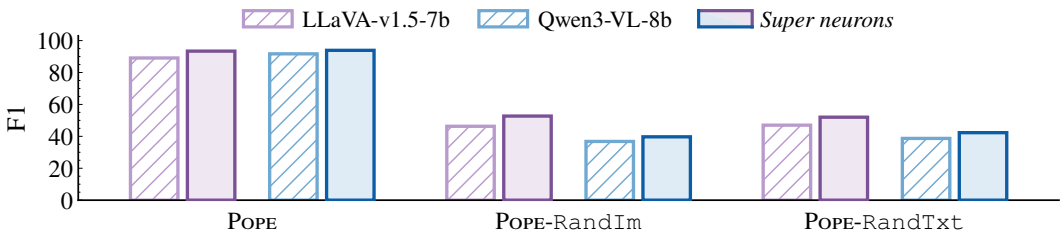



Figure 3: Performance of LLaVA-v1.5-7b and Qwen3-VL-8b-Instruct (hashed) and their SNs (solid) on image- and text-shuffled POPE datasets. F1 collapses in both adversarial settings, suggesting that *super neurons* rely on both vision and text, but *not* on spurious statistics from the model or the data.

**Implementation details.** We use LLaVA-v1.5 (Liu et al., 2023) and the Qwen3-VL-Instruct (Qwen Team, 2025) family of models across our manuscript. We use the checkpoints from their official huggingface repositories. We detail the configurations of the VLMs in appendix A.3. Our algorithm is fast and memory efficient, allowing us to use a single NVIDIA RTX A6000 GPU for all our *super neurons* experiments. The extraction of *super neurons* is training-free and simply requires the collection of hidden states from LLM of the VLM. We default our setup to $k = 10$ *super neurons*. For models, we use 128 max. generated tokens, set temperature to 0, use 1 beam, and do not use stop strings.

Table 1: Spatial understanding of Qwen3-VL-8b-Instruct. *Super neurons* obtain remarkable F1.

| Dataset | Model | SNs |
|---|---|---|
| INSTAORDER (DEPTH) | 18.1 | **69.6**(+51.5) |
| INSTAORDER (OCC.) | 2.8 | **33.7**(+30.9) |
| CLEVR | 74.4 | **93.4**(+19.0) |

### 4.2 EXISTENCE AND CAPABILITIES OF *Super neurons*

We run algorithm 1 for each dataset in our suite. For all experiments, we extract *super neurons* on the probing set and evaluate them on the validation set.

***Super neurons* exist.** An ensemble of 10 task-specific expert neurons outperforms the base models' predictions made from billions of units on CVQA (fig. 2). For both models, *this represents about 0.007% of the total activations*. Detailed results are in appendix C.

Table 2: Out-of-distribution performance of *super neurons*.

| Method | Probe | Val. | Acc. | F1 |
|---|---|---|---|---|
| LLaVA-v1.5-7b | – | VOC | 95.0 | 95.2 |
| *Super neurons* | POPE | VOC | **96.8**(+1.8) | **96.9**(+1.7) |

***Super neurons* are strong classifiers.** We observe that *super neurons* are effective on saturated benchmarks such as POPE (fig. 2). *Super neurons* also challenge the prevailing view that VLMs lack robust 3D understanding from 2D scenes, obtaining strong results on spatial understanding benchmarks (table 1). Although the accuracy of Qwen3-VL-8b-Instruct diminishes from 84.6% to 68.4% on instance-wise occlusion prediction (fig. 2), *super neurons* achieve higher F1 scores than the base models on all spatial datasets (table 1). We found that the base model's accuracy is close to the proportion of "no" labels in INSTAORDER (OCC.), explaining its high accuracy but poor 2.8 F1 score. Since both base models struggle on spatial understanding datasets but not their *super neurons*, we believe they could be strong alternatives for improving performance on out-of-distribution data at a low annotation and compute cost. As a matter of fact, *super neurons* still outperform the base models on VIZWIZ although less than 1,000 probing samples are available. *Super neurons* also comfortably outperform the base models on MCQ datasets (fig. 2).

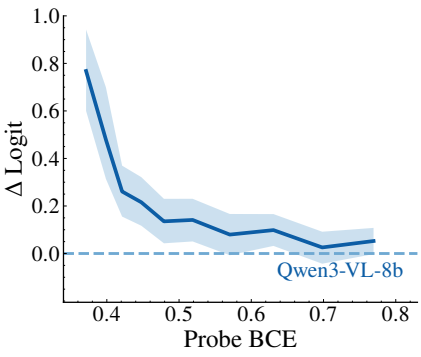



Figure 4: Effect of patching single layer-35 units of Qwen3-VL-8b against their probe BCE on POPE.

***Super neurons* show no evidence of overfitting.** To validate that *super neurons* are not overfitting artifacts, we conduct a simple control experiment on POPE by adversarially shuffling its images (POPE-RandIm) and prompts (POPE-RandTxt). If *super neurons* were to rely on spurious biases or exploit dataset statistics, one would expect to see a high score on these datasets too. However, fig. 3 shows that *super neurons*' performance collapses on both POPE-RandIm and POPE-RandTxt, suggesting that they do not rely on spurious signals. It also confirms that *super neurons* rely on both image and text representations for their predictions.

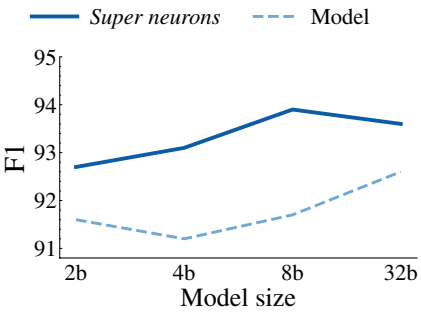



Figure 5: *Super neurons*' F1 with respect to the size of the Qwen-VL-Instruct model they are elected from.

**Out-of-distribution evaluation.** We investigate out-of-distribution capabilities of *super neurons* by probing them on the probing set of POPE and evaluating them on a POPE-style benchmark built from VOC images (Everingham et al., 2015). As shown in table 2, *super neurons* still outperform the base model, indicating a degree of performance transfer beyond the dataset they are probed on.

***Super neurons* play a greater causal role.** To test whether *super neurons* causally contribute to the model's output, we patch individual neurons of Qwen3-VL-8b-Instruct at the last prompt token on POPE and read the resulting change in the yes–no logit margin. The activation is shifted by a fixed standardized amount away from ground truth. Figure 4 shows that our selection criterion tracks causal influence: neurons in the lowest BCE decile move the margin roughly 15× more than those in the highest. This trend appears at layer 25 and strengthens with depth, which is where most *super neurons* are located (appendix D).

Table 3: Performance of few-shot prompting and *super neurons* on POPE. Best shot is underlined.

| Num. Shots | LLaVA-v1.5-7b | | Qwen3-VL-8b-Instruct | |
|---|---|---|---|---|
| | Acc. | F1 | Acc. | F1 |
| 0 | 88.7 | 89.1 | 92.0 | 91.7 |
| 1 | 90.1 (+1.4) | 90.0 (+0.9) | 92.8 (+0.8) | 92.6 (+0.9) |
| 3 | 88.4 (-0.3) | 88.0 (-1.1) | 92.2 (+0.2) | 92.0 (+0.3) |
| 5 | 88.4 (-0.3) | 88.0 (-1.1) | 91.8 (-0.2) | 91.4 (-0.3) |
| SNs | **93.2** (+4.5) | **93.4** (+4.3) | **93.8** (+1.8) | **93.9** (+2.2) |

**Scaling properties.** We scale the Qwen3-VL-Instruct model family from 2 to 32 billion parameters in fig. 5. *Super neurons* always outperform the model, no matter the size. We observe stable scaling up to 8b parameters. While the 32b model seems to show slightly less performance when $k = 10$, simply increasing $k$ naturally recovers the gap. Setting $k = 100$ yields 94.2 F1, smoothly surpassing the 8b model. We believe that this might stem from the fact that models up to 8b parameters were distilled from the 32b model during post-training (Qwen Team, 2025). We conclude that *super neurons* show natural, desirable scaling properties.

**Qualitative results.** In fig. 1, we display an example from the 10 selected *super neurons* of LLaVA-v1.5-7b on the validation set of POPE. *Super neurons* are more often correct and more confident than the model, even on tricky cases. We display more results in appendix E.

### 4.3 COMPARISON WITH BASELINES

**Few-shot prompting.** We first compare our *super neurons* to few-shot prompting (Agarwal et al., 2024), since it is a well-known gradient-free regime that matches our approach's nature well. We report results on POPE in table 3. Detailed results are in tables 10 and 11, in appendix C.3.

As the number of shots used to prompt the model increases, performance degrades. While surprising, this is consistent with previously reported results on the topic (Mitra et al., 2025). Nonetheless, *super neurons* consistently perform better than few-shot prompting.

**Sparse attention vectors.** We also compare our *super neurons* to sparse attention vectors (SAVs), since this approach also leverages pretrained features of the LLM for classification purposes (Mitra

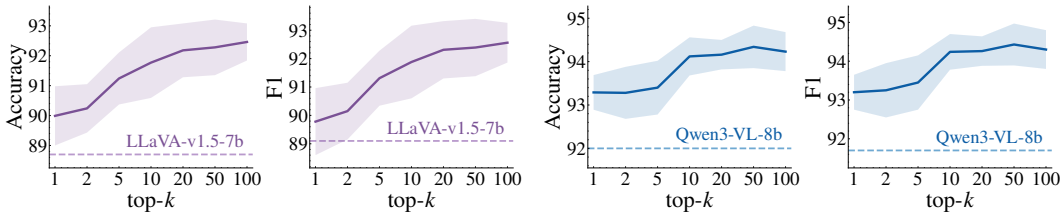



Figure 7: Performance on the validation set of POPE with respect to the number of *super neurons* $k$ used for prediction. All values outperform their respective base models. The shaded area represents the 95% confidence interval, computed across 5 different probing sets.

et al., 2025). We report the full benchmarks in tables 12 and 13, in appendix C.4. We surpass SAVs across almost all datasets, on both models. We also note that while SAVs consistently improve accuracy, it is not always the case with F1. *Super neurons*, on the other hand, are more reliable.

To assess the shift from vector to scalar representations induced from SAVs to *super neurons*, we plot the performance of both approaches with respect to their effective parameter count on POPE (fig. 6). *Super neurons* achieve a new Pareto frontier, obtaining better performance than SAVs while relying on two orders of magnitude fewer parameters. We believe this arises from *super neurons* searching over a much larger scalar-level space than SAVs. Specifically, *super neurons* consider $L \times D$ candidate scalars, whereas SAVs consider $L \times A_h$ attention heads, with $A_h \ll D$.

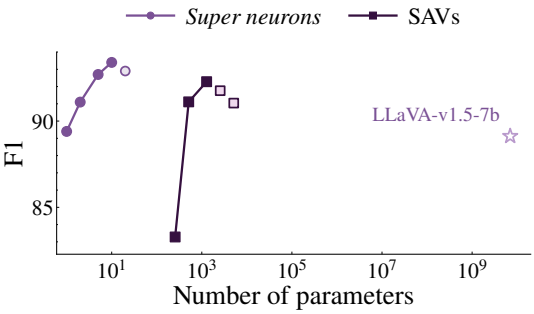



Figure 6: Performance of *super neurons* and SAVs with respect to their number of parameters on POPE. SNs achieve a new Pareto frontier.

### 4.4 ABLATION EXPERIMENTS

**Maximizing the efficiency of *super neurons*.** We vary the number of selected units for ensemble inference using our strategy in fig. 7. While a *single unit* is able to outperform each of their respective models, increasing the number of *super neurons* provides consistent gains.

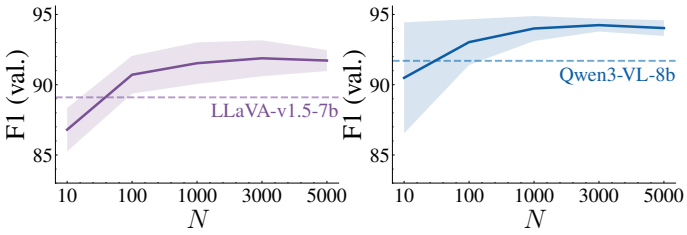



Figure 8: *Super neurons*' F1 with respect to the size $N$ of the probing set on POPE. The shaded area represents the 95% confidence interval, computed across 5 different probing sets.

In fig. 8, we show that *super neurons* can outperform the model within the 100 samples region. However, we report the best gain of performance at $N = 3{,}000$ samples.

### 4.5 APPLICATIONS OF *Super neurons*

While *super neurons* are probed and validated on CVQA data, their applications extend beyond improvements on categorical benchmarks. We explore some of their potential in this section.

**Extreme early exiting.** We report the full prefill-decode analysis of our *super neurons* with respect to the base model they are extracted from in table 4. Since *super neurons* rely on scalar-level representations extracted from the forward pass of the prefill tokens in the LLM, their predictions do not emit a single token. We call this property *extreme early exiting*. This translates into substantial inference speedup of 6.35× and 27.8× over the base LLaVA-v1.5-7b and Qwen3-VL-8b-Instruct, respectively. Extreme early exiting suggests that task-relevant representations are already internalized in the LLM of the VLM, as soon as the prefill is encoded.

Table 4: Prefill-decode analysis on POPE ($N = 3{,}000$). AR indicates "autoregressive" prediction.

| Method | AR | Prefill (ms.) | Decode (ms.) | Total (ms.) | Tokens | Speedup |
|---|---|---|---|---|---|---|
| LLaVA-v1.5-7b | ✓ | 113.21 | 605.20 | 718.41 | 23.63 | 1.00× |
| *Super neurons* | ✗ | 113.21 | 0.00 | **113.21** | 0 | **6.35**× |
| Qwen3-VL-8b-Instruct | ✓ | 105.22 | 2816.62 | 2921.84 | 80.72 | 1.00× |
| *Super neurons* | ✗ | 105.22 | 0.00 | **105.22** | 0 | **27.80**× |

Table 5: Performance of LLaVA-v1.5-7b after being finetuned on INSTAORDER (OCC.).

| | INSTAORDER (OCC.) | | POPE | | | |
|---|---|---|---|---|---|---|
| Method | Acc. | F1 | Acc. | F1 | GPUs | GPU-hours (↓) |
| Base | 24.6 | 22.9 | 88.7 | 89.1 | N/A | N/A |
| SFT | **74.8**(+50.2) | 36.6(+13.7) | 59.1(-29.6) | 62.2(-26.9) | 8 | 2.387 |
| LoRA | 65.4(+40.8) | 29.5(+6.6) | 51.6(-37.1) | 67.8(-21.3) | 4 | 0.369 |
| SNs | 31.1(+6.5) | **42.0**(+19.1) | **82.6**(-6.1) | **85.2**(-3.9) | **1** | **0.183** |

**Finetuning.** We hypothesize that finetuning *super neurons* might improve the model's performance on a target task while maintaining some of its generalist behavior on others. Thus, we finetune the units that contribute to the 10 best *super neurons* of LLaVA-v1.5-7b on INSTAORDER (OCC.), since VLMs struggle most on this dataset. We compare our approach to LoRA (Hu et al., 2022) and full finetuning of the LLM (SFT). We train all methods for a single epoch using a balanced low-budget training set of 3,000 samples. We detail the full experimental setup in appendix A.4. Our *super neurons* finetuning setup runs on a single GPU, in less than 10 minutes.

Table 6: Performance of LLaVA-v1.5-7b on MM-VET after being finetuned on INSTAORDER (OCC.).

| Method | Score (↑) |
|---|---|
| Base | 28.8 |
| SFT | 4.6(-24.2) |
| LoRA | 6.2(-22.6) |
| SNs | **30.4**(+1.6) |

Results in table 5 indicate that *super neurons* finetuning brings the most F1 gains while maintaining general performance on POPE, after altering the weights. Both LoRA and SFT greatly improve accuracy, but end up exploiting the natural data imbalance of the task, as their low F1 testifies, although the training set is balanced. Moreover, these baselines specialize the model to the target task, making their predictions on POPE slightly better than chance. This is clearly assessed with evaluation on MM-VET (Yu et al., 2024), a dataset requiring open-ended answers. In table 6, we show that *super neurons* have a positive impact on the final score, while LoRA and SFT collapse. This improvement mostly comes from spatial understanding transfer (table 16, appendix C.8). We provide more details on these finetuning experiments in appendix C.8. Under this setting, these results suggest that *super neurons* could become a strong finetuning option.

## 5 CONCLUSION

**Takeaway.** Using a small budget of categorical VQA samples, we uncover the existence of task-specific expert units in VLMs. Ensembling as few as 10 of these *super neurons* suffices to outperform the full billion-parameter model from which they are extracted. These results invite us to reconsider the information locality and sparsity in large pretrained networks. Performance-wise, *super neurons* compare favorably to few-shot prompting and SAVs. Application-wise, *super neurons* bypass the autoregressive process of the LLM, enabling extreme early exiting and substantially reducing runtime. Furthermore, finetuning 10 *super neurons* is enough to significantly improve the performance of the resulting model on the target task, while maintaining its general capabilities on other tasks.

**Limitations.** *Super neurons* cannot be extracted from closed-weight models since they require at least partial access to the parameters of the model. We do not investigate the reason this phenomenon emerges in the first place and leave it as future work.

### AI USE STATEMENT

In this work, we used generative AI tools (GPT-5.6-Terra and Fable 5.1) for searching related works, verifying implementations, debugging, suggesting minor optimization schemes, writing bash scripts managing experimental setups, polishing the manuscript. Specifically, generative AI was used to design fig. 2 from numbers manually extracted from the experiments.

We have not used generative AI tools for ideation (whether in the core idea of the paper, or any kind of experimental ideation), and writing code for our paper without a skeleton or basic coding specifications.

We have reviewed all AI-assisted work. Specifically, we manually read all the references agents recommended to add to this work, we proofread all the bash scripts used to setup the experiments, we never blindly copy-pasted manuscript suggestions from the agents, but instead naturally integrated some of the suggestions of the software by rewriting it ourselves.

We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

### ETHICS STATEMENT

We fully adhere to the Code of Ethics of ICLR.

We do not believe our work raises concerns regarding any of the items mentioned in it. Our work does not involve human subjects, we do not release data and simply use publicly available datasets that are well-known in our research community, we declare no conflict of interest, nor sponsorship, Our work is not used nor intended to support deceptive, harmful, or malicious uses.

### REPRODUCIBILITY STATEMENT

Our work can be easily replicated using the pseudo-code available in algorithm 1. We thoroughly detail each of the steps of our method in section 3, which should be sufficient for readers to reproduce our approach.

Nonetheless, we will release our code after acceptance to ICLR 2027, allowing clear and transparent reproduction, but also modification for any reader. The code base relies on public common libraries such as pytorch and huggingface and does not need custom or private repositories to be run.

## REFERENCES

Rishabh Agarwal, Avi Singh, Lei Zhang, Bernd Bohnet, Luis Rosias, Stephanie C. Y. Chan, Biao Zhang, Ankesh Anand, Zaheer Abbas, Azade Nova, John D. Co-Reyes, Eric Chu, Feryal M. P. Behbahani, Aleksandra Faust, and Hugo Larochelle. Many-shot in-context learning. In *NeurIPS*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/8cb564df771e9eacbfe9d72bd46a24a9-Abstract-Conference.html. 7

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey A. Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, Julian Eisenschlos, Rishabh Kabra, Matthias Bauer, Matko Bosnjak, Xi Chen, Matthias Minderer, Paul Voigtlaender, Ioana Bica, Ivana Balazevic, Joan Puigcerver, Pinelopi Papalampidi, Olivier J. Hénaff, Xi Xiong, Radu Soricut, Jeremiah Harmsen, and Xiaohua Zhai. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024. URL https://doi.org/10.48550/ARXIV.2407.07726. 1, 3

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *CVPR*, pages 24185–24198, 2024. URL https://doi.org/10.48550/ARXIV.2312.14238. 1

An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. Spatialrgpt: Grounded spatial reasoning in vision-language models. In *NeurIPS*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/f38cb4cf9a5eaa92b3cfa481832719c6-Abstract-Conference.html. 15

Mark Everingham, S. M. Ali Eslami, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, pages 98–136, 2015. URL https://doi.org/10.1007/S11263-014-0733-5. 7, 15

Danna Gurari, Qing Li, Abigale J. Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P. Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *CPVR*, pages 3608–3617, 2018. URL https://openaccess.thecvf.com/content_cvpr_2018/papers/Gurari_VizWiz_Grand_Challenge_CVPR_2018_paper.pdf. 5, 15, 16

Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. Finding neurons in a haystack: Case studies with sparse probing. *TMLR*, 2023. URL https://openreview.net/forum?id=JYs1R9IMJr. 3

Alberto Hojel, Yutong Bai, Trevor Darrell, Amir Globerson, and Amir Bar. Finding visual task vectors. In *ECCV*, pages 257–273, 2025. URL https://doi.org/10.1007/978-3-031-72775-7_15. 3

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *ICLR*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9. 2, 9, 17

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *ICLR*, 2024. URL https://openreview.net/forum?id=F76bwRSLeK. 3

Ting Jiang, Jian Jiao, Shaohan Huang, Zihan Zhang, Deqing Wang, Fuzhen Zhuang, Furu Wei, Haizhen Huang, Denvy Deng, and Qi Zhang. Promptbert: Improving bert sentence embeddings with prompts. In *EMNLP*, pages 8826–8837, 2022. URL https://doi.org/10.18653/V1/2022.EMNLP-MAIN.603. 3

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross B. Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, pages 1988–1997, 2017. URL https://doi.org/10.1109/CVPR.2017.215. 5, 15

Seil Kang, Jinyeong Kim, Junhyeok Kim, and Seong Jae Hwang. See what you are told: Visual attention sink in large multimodal models. In *ICLR*, 2025a. URL https://openreview.net/forum?id=7uDI7w5RQA. 3

Seil Kang, Jinyeong Kim, Junhyeok Kim, and Seong Jae Hwang. Your large vision-language model only needs a few attention heads for visual grounding. In *CVPR*, pages 9339–9350, 2025b. URL https://openaccess.thecvf.com/content/CVPR2025/html/Kang_Your_Large_Vision-Language_Model_Only_Needs_A_Few_Attention_Heads_CVPR_2025_paper.html. 1

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Paul Foster, Pannag R. Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. In *CoRL*, pages 2679–2713, 2024. URL https://proceedings.mlr.press/v270/kim25c.html. 1, 15

Quoc V. Le, Marc'Aurelio Ranzato, Rajat Monga, Matthieu Devin, Kai Chen, Greg Corrado, Jeffrey Dean, and Andrew Y. Ng. Building high-level features using large scale unsupervised learning. In *ICML*, 2012. URL http://icml.cc/2012/papers/73.pdf. 3

Hyunmin Lee and Jaesik Park. Instance-wise occlusion and depth orders in natural scenes. In *CVPR*, pages 21178–21189, 2022. URL https://doi.org/10.1109/CVPR52688.2022.02053. 15

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *EMNLP (1)*, pages 3045–3059, 2021. URL https://doi.org/10.18653/V1/2021.EMNLP-MAIN.243. 3

Fuhao Li, Wenxuan Song, Han Zhao, Jingbo Wang, Pengxiang Ding, Donglin Wang, Long Zeng, and Haoang Li. Spatial forcing: Implicit spatial representation alignment for vision-language-action model. In *ICLR*, volume 2026, pages 132324–132345, 2026. URL https://openreview.net/forum?id=euMVC1DO4k. 15

Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, pages 19730–19742, 2023a. URL https://openreview.net/forum?id=KU9UojoX7U. 3

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *EMNLP*, pages 292–305, 2023b. URL https://doi.org/10.18653/V1/2023.EMNLP-MAIN.20. 5, 15, 16

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/6dcf277ea32ce3288914faf369fe6de0-Abstract-Conference.html. 1, 3, 6

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *CVPR*, pages 26286–26296, 2024. URL https://doi.org/10.1109/CVPR52733.2024.02484. 1

Zhijian Liu, Ligeng Zhu, Baifeng Shi, Zhuoyang Zhang, Yuming Lou, Shang Yang, Haocheng Xi, Shiyi Cao, Yuxian Gu, Dacheng Li, Xiuyu Li, Haotian Tang, Yunhao Fang, Yukang Chen, Cheng-Yu Hsieh, De-An Huang, An-Chieh Cheng, Jinyi Hu, Sifei Liu, Ranjay Krishna, Pavlo Molchanov, Jan Kautz, Hongxu Yin, Song Han, and Yao Lu. Nvila: Efficient frontier visual language models. In *CVPR*, pages 4122–4134, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Liu_NVILA_Efficient_Frontier_Visual_Language_Models_CVPR_2025_paper.html. 1, 3

Llama Team. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://doi.org/10.48550/ARXIV.2407.21783. 1, 3

Haoran Lou, Chunxiao Fan, Ziyan Liu, Yuexin Wu, and Xinliang Wang. Llava-sp: Enhancing visual representation with visual spatial tokens for mllms. In *ICCV*, pages 22004–22024, 2025. URL https://doi.org/10.1109/ICCV51701.2025.02044. 3

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *NeurIPS*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/11332b6b6cf4485b84afadb1352d3a9a-Abstract-Conference.html. 5, 15, 16

Chancharik Mitra, Brandon Huang, Tianning Chai, Zhiqiu Lin, Assaf Arbelle, Rogério Feris, Leonid Karlinsky, Trevor Darrell, Deva Ramanan, and Roei Herzig. Enhancing few-shot vision-language classification with large multimodal model features. In *ICCV*, pages 2760–2772, 2025. URL https://doi.org/10.1109/ICCV51701.2025.00265. 1, 3, 7, 16, 21

Pierre Musacchio, Hyunmin Lee, and Jaesik Park. Holistic order prediction in natural scenes. In *NeurIPS*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/150f35763dc51bfc269690d36a5a7c88-Abstract-Conference.html. 5, 15, 16

Tuomas P. Oikarinen and Tsui-Wei Weng. Clip-dissect: Automatic description of neuron representations in deep vision networks. In *ICLR*, 2023. URL https://openreview.net/forum?id=iPWiwWHc1V. 3

Ziqi Pang, Ziyang Xie, Yunze Man, and Yu-Xiong Wang. Frozen transformers in language models are effective visual encoder layers. In *ICLR*, 2024. URL https://openreview.net/forum?id=t0FI3Q66K5. 3

Qwen Team. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025. URL https://doi.org/10.48550/ARXIV.2511.21631. 1, 3, 6, 7, 23

Alec Radford, Rafal Józefowicz, and Ilya Sutskever. Learning to generate reviews and discovering sentiment. *arXiv preprint arXiv:1704.01444*, 2017. URL http://arxiv.org/abs/1704.01444. 3

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763, 2021. URL http://proceedings.mlr.press/v139/radford21a.html. 3

Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-okvqa: A benchmark for visual question answering using world knowledge. In *ECCV*, pages 146–162, 2022. URL https://doi.org/10.1007/978-3-031-20074-8_9. 5, 15, 16

Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. Layer by layer: Uncovering hidden representations in language models. In *ICML*, 2025. URL https://proceedings.mlr.press/v267/skean25a.html. 1, 3

Zineng Tang, Long Lian, Seun Eisape, Xudong Wang, Roei Herzig, Adam Yala, Alane Suhr, Trevor Darrell, and David M. Chan. Tulip: Towards unified language-image pretraining. *arXiv preprint arXiv:2503.15485*, 2025. URL https://doi.org/10.48550/ARXIV.2503.15485. 3

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L. Turner, Callum McDougall, Monte MacDiarmid, Alex Tamkin, Esin Durmus, Tristan Hume, Francesco Mosconi, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam S. Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *arXiv preprint arXiv:2605.29358*, 2026. URL https://doi.org/10.48550/ARXIV.2605.29358. 3

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *CVPR*, pages 9568–9578, 2024. URL https://doi.org/10.1109/CVPR52733.2024.00914. 3

Michael Tschannen, Alexey A. Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier J. Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025. URL https://doi.org/10.48550/ARXIV.2502.14786. 3

Xiaozhi Wang, Kaiyue Wen, Zhengyan Zhang, Lei Hou, Zhiyuan Liu, and Juanzi Li. Finding skill neurons in pre-trained transformer-based language models. In *EMNLP*, pages 11132–11152, 2022. URL https://doi.org/10.18653/V1/2022.EMNLP-MAIN.765. 3

Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. In *ICML*, pages 57730–57754, 2024. URL https://proceedings.mlr.press/v235/yu24o.html. 9, 15, 18, 25

Zhuoran Yu and Yong Jae Lee. How multimodal llms solve image tasks: A lens on visual grounding, task reasoning, and answer decoding. In *COLM*, 2025. URL https://openreview.net/forum?id=kjNJYWvfPA. 1, 3

Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. In *EMNLP (Demos)*, pages 543–553, 2023. URL https://doi.org/10.18653/V1/2023.EMNLP-DEMO.49. 1, 3

# APPENDICES

## SUPER NEURONS ANSWER BEFORE VLMS SPEAK

**Contents**


**A Additional experimental details** **15**
- A.1 Datasets . . . 15
- A.2 Prompts . . . 15
- A.3 Baselines . . . 16
- A.4 Finetuning . . . 17

**B Agreement rate** **18**

**C Additional quantitative results** **18**
- C.1 Evaluation metrics . . . 18
- C.2 Main benchmark discussions . . . 19
- C.3 Comparison with few-shot prompting . . . 21
- C.4 Comparison with SAVs . . . 22
- C.5 Decorrelation analysis . . . 23
- C.6 Scaling *Super neurons* . . . 23
- C.7 Analysis of the number of *Super neurons* . . . 24
- C.8 Finetuning results . . . 25

**D Additional causal experiments** **26**

**E Qualitative results** **27**

# A Additional experimental details

## A.1 Datasets

**Main experiments.** We describe the datasets along with the prompts used for probing and validation in more detail in this section.

**POPE.** POPE is a dataset designed to evaluate the degree of hallucinations of VLMs. Built on top of COCO, it consists of 2.9K VQA that relate to the presence or absence of objects in images, *i.e.*, to visual hallucinations (Li et al., 2023b). It officially has no training set, but since it is constructed from COCO images, we simply use the COCO training images to build one.

**INSTAORDER (OCC.).** INSTAORDER also stems from COCO and consists of instance-wise ordering annotations (Lee and Park, 2022). Recent work converted this dataset into a VQA format and observed that LLaVA-v1.5-7b struggles to answer such geometry-related prompts (Musacchio et al., 2025). This split of the dataset tests the models' understanding of instance-wise occlusions.

**INSTAORDER (DEPTH).** Second split of INSTAORDER, the questions contained in this challenge test the models' abilities to reason about instance-wise depth order. We believe this dataset is interesting to benchmark since it aims at scoring important spatial understanding capabilities that could then be re-used via vision-language action models for robotic manipulations (Li et al., 2026; Cheng et al., 2024; Kim et al., 2024).

**VIZWIZ.** Created by people with visual impairments, VIZWIZ is a general benchmark aiming at testing models' capacities to reason under constrained visual inputs. We filter the dataset on "yes"-"no" answers and only keep such questions for both the probing and validation sets (Gurari et al., 2018). Fewer than 1K samples meet this criterion, so we use all of them for probing.

**CLEVR.** To test our method's generalization capabilities on synthetic data, we select CLEVR. This dataset evaluates the spatial understanding of the model through questions that span from counting to positional understanding. We filter this dataset to only retain classification questions for probing and validation (Johnson et al., 2017).

**A-OKVQA.** A-OKVQA is an MCQ dataset with four answer choices consisting of general visual world knowledge. Questions require commonsense reasoning grounded in the scene in order to be accurately answered (Schwenk et al., 2022).

**SCIENCEQA.** We benchmark our approach on this high school science dataset. The SCIENCEQA dataset contains 26 topics, 127 categories, and 379 skills (Lu et al., 2022). We convert this MCQ data to a series of binary visual question answer and perform inference in the same way as on the other datasets.

**Additional experiments.** For additional experiments, we also use the following datasets.

**VOC.** The VOC dataset is a seminal computer vision dataset usually used for object classification or segmentation (Everingham et al., 2015). In our work, we leverage images and classes from VOC and convert them to a POPE-like format using the official POPE dataset converter.

**MM-VET.** For evaluating the capabilities of our finetuned models, we leverage MM-VET (Yu et al., 2024). It is an open-ended VQA dataset, requiring the model to produce sentence-level outputs. It measures common understanding, OCR and spatial reasoning.

## A.2 Prompts

For CLEVR and VIZWIZ, we do not apply any prompt template and simply filter the dataset to retain only "yes"-"no" VQA in both the training and validation sets. We prompt the model with the filtered samples. For the other datasets, we report the prompt templates we use in table 7.

Table 7: Prompt templates used during our experiments on the CVQA datasets. {} indicates variables, `obj` indicates the object category, `loc` indicates the location of the object in a bounding box format. For MCQ data, `q` indicates the original question and $a_i$ indicates the $i$-th MCQ answer.

| Dataset | Reference | Prompt |
|---|---|---|
| POPE | Li et al. (2023b) | `Is there a {obj} in the image?` |
| INSTAORDER (OCC.) | Musacchio et al. (2025) | `Is the {obj} at position {loc} occluding the {obj} at position {loc}? Answer the question using a single word.` |
| INSTAORDER (DEPTH) | Musacchio et al. (2025) | `Is the {obj} at position {loc} in front of the {obj} at position {loc} from the viewer's perspective? Answer the question using a single word.` |
| A-OKVQA | Schwenk et al. (2022) | {q} `Is it` $\{a_i\}$`? Answer with yes or no.` |
| SCIENCEQA | Lu et al. (2022) | {q} `Is it` $\{a_i\}$`? Answer with yes or no.` |

Note that for INSTAORDER, the location of the object is only specified if there are multiple same object categories in the image. This allows disambiguation of the target object. Otherwise, only the object category is specified.

### A.3 BASELINES

**Model configurations.** We report the configurations of LLaVA-v1.5-7b and the Qwen-VL-Instruct family used in our experiments in table 8 of appendix A.3. We default to the huggingface implementations and checkpoints[1,2] for all our experiments except for finetuning LLaVA-v1.5-7b. In this case, we opt for the official LLaVA repository which contains reproducible training scripts. We do not alter these configurations unless specified otherwise in the paper.

**Few-shot prompting.** For few-shot prompting, we randomly sample 5 samples from the probing set. We enforce 3 positive and 2 negative classes. During validation, we simply prepend the shots to the actual image and prompt of interest. We compare these results to the $k = 10$ best extracted *super neurons* on each dataset.

**SAVs.** We use the default SAVs configuration[3], as specified in their work (Mitra et al., 2025). Namely, for each class in the dataset, we compute the centroids attention vectors across the $N = 20$ positive and negative samples. We then compute their accuracy and select the top-$k$, with $k = 20$ best heads. We compare these results to the $k = 10$ best extracted *super neurons* on each dataset. SAVs (Mitra et al., 2025) are also benchmarked on VIZWIZ (Gurari et al., 2018), but the training and validation set are constructed based on the "answerable"-"non-answerable" fields, which most of the time does not reflect the intent of the question well. Instead, we select the "yes"-"no" questions of VIZWIZ only and evaluate both approaches on it.

**GPU usage.** As reported in the main manuscript, all experiments were conducted on NVIDIA RTX A6000 GPUs. We make a single exception when we run inference on Qwen3-VL-32b-Instruct, since it generates a larger memory footprint. In this case, we use a single NVIDIA A100 GPU.

[1] https://huggingface.co/llava-hf/llava-1.5-7b-hf
[2] https://huggingface.co/collections/Qwen/qwen3-vl
[3] https://github.com/chancharikmitra/SAVs

Table 8: Configurations of the VLMs we use throughout our work.

(a) LLaVA-v1.5-7b.

| | |
|---|---|
| model_type | llava |
| dtype | float16 |
| head_dim | 128 |
| hidden_act | silu |
| hidden_size | 4096 |
| model_type | llama |
| num_attention_heads | 32 |
| num_hidden_layers | 32 |
| num_key_value_heads | 32 |
| vocab_size | 32064 |

(b) Qwen3-VL-2b-Instruct.

| | |
|---|---|
| model_type | qwen3_vl |
| dtype | bfloat16 |
| head_dim | 128 |
| hidden_act | silu |
| hidden_size | 2048 |
| model_type | qwen3_vl_text |
| num_attention_heads | 16 |
| num_hidden_layers | 28 |
| num_key_value_heads | 8 |
| vocab_size | 151936 |

(c) Qwen3-VL-4b-Instruct.

| | |
|---|---|
| model_type | qwen3_vl |
| dtype | float16 |
| head_dim | 128 |
| hidden_act | silu |
| hidden_size | 2560 |
| model_type | qwen3_vl_text |
| num_attention_heads | 32 |
| num_hidden_layers | 36 |
| num_key_value_heads | 8 |
| vocab_size | 151936 |

(d) Qwen3-VL-8b-Instruct.

| | |
|---|---|
| model_type | qwen3_vl |
| dtype | bfloat16 |
| head_dim | 128 |
| hidden_act | silu |
| hidden_size | 4096 |
| model_type | qwen3_vl_text |
| num_attention_heads | 32 |
| num_hidden_layers | 36 |
| num_key_value_heads | 8 |
| vocab_size | 151936 |

(e) Qwen3-VL-32b-Instruct.

| | |
|---|---|
| model_type | qwen3_vl |
| dtype | bfloat16 |
| head_dim | 128 |
| hidden_act | silu |
| hidden_size | 5120 |
| model_type | qwen3_vl_text |
| num_attention_heads | 64 |
| num_hidden_layers | 64 |
| num_key_value_heads | 8 |
| vocab_size | 151936 |

Probing the data to discover *super neurons* is fast and memory-friendly. We use a single GPU to extract them for all datasets and all models, which usually takes less than 30 minutes. It is easy to parallelize the probing strategy on 8 GPUs, which reduces the probing time to approximately 3 minutes.

### A.4 Finetuning

**Implementation.** For all finetuning experiments, we use the official repository of LLaVA-v1.5-7b[4]. We use the default LoRA task training for training LoRA (Hu et al., 2022) models and the default supervised finetuning trainer when training for SFT. We do not alter the default hyperparameters. We keep a global batch size of 64 for both methods and train on 4 GPUs for LoRA and 8 GPUs for SFT.

For finetuning *super neurons*, we freeze the entire model, use the indices of the *super neurons* and only finetune the neurons that contribute to the best $k = 10$ units, including those coming from the residual stream. We use the same next token prediction objective as for LoRA and SFT finetuning, use the AdamW optimizer, a batch size of 8 and a learning rate of 1e-5. We set weight decay to 0

[4] https://github.com/haotian-liu/LLaVA

Table 9: Agreement rate between LLaVA-v1.5-7b and its *super neurons* on POPE. We vary the size of the ensemble $k$ to understand how the agreement evolves.

| $k$ | $\rho_{\mathcal{S}^\star}$ |
|---|---|
| 1 | 93.8 |
| 2 | 94.5 |
| 5 | 91.2 |
| 10 | 90.7 |
| 20 | 90.3 |
| 50 | 90.8 |
| 100 | 91.2 |

and do not use any scheduler. Generally speaking, we do not leverage any special trick and keep finetuning as plain as possible to show that *super neurons* finetuning can work out of the box, without a lot of hyperparameters.

We train all our models on NVIDIA RTX A6000 GPUs. All setups use a training set of 3,000 samples.

**Evaluation on MM-VET.** We evaluate all our trained models on MM-VET (Yu et al., 2024) using the official online evaluation website[5]. This evaluation platform uses GPT 4.1 as a judge.

## B AGREEMENT RATE

**Definition.** To assess the differences between the base model and $k \in \mathbb{N}^*$ *super neurons*, we also propose a simple agreement rate score $\rho$, defined as:

$$\rho_{\mathcal{S}} = \frac{1}{NK} \sum^{N} \sum^{K} \mathbb{1}\left(\hat{\mathbf{Y}}_{\mathcal{S},n} = \hat{\mathbf{Y}}_n\right) \in [0, 1]. \tag{6}$$

Here, $\hat{\mathbf{Y}}_{\mathcal{S},n}$ denotes the predictions from a subset of neurons $\mathcal{N}$ of the LLM. Intuitively, a high $\rho_{\mathcal{S}}$ indicates that the predictions of *super neurons* match the predictions of the base model while a lower value indicates that the predictions made from *super neurons* diverge from it.

**Empirical analysis.** We compute the agreement rate between LLaVA-v1.5-7b and its *super neurons* for varying $k$'s on POPE in table 9. We observe that using $k = 10$, our default value for the ensemble size of *super neurons* across the manuscript results in one of the lowest agreement rate. It is understandable that *super neurons* must disagree with the model to a certain extent, in order to be able to outperform it.

## C ADDITIONAL QUANTITATIVE RESULTS

In this section, we provide the full results of the experiments led and reported in section 4 of the main manuscript.

### C.1 EVALUATION METRICS

We evaluate vanilla models and *super neurons* using traditional accuracy, precision, recall and F1, which are defined as follows:

$$\text{Accuracy} = \frac{1}{N} \sum_n \mathbb{1}\left(\hat{\mathbf{Y}}_n = \mathbf{Y}_n\right), \qquad \text{Precision} = \frac{\sum_n \mathbb{1}\left(\hat{\mathbf{Y}}_n = 1 \wedge \mathbf{Y}_n = 1\right)}{\sum_n \mathbb{1}\left(\hat{\mathbf{Y}}_n = 1\right)},$$

$$\text{Recall} = \frac{\sum_n \mathbb{1}\left(\hat{\mathbf{Y}}_n = 1 \wedge \mathbf{Y}_n = 1\right)}{\sum_n \mathbb{1}(\mathbf{Y}_n = 1)}, \qquad \text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.$$

[5] https://huggingface.co/spaces/whyu/MM-Vet_Evaluator

Table 10: Comparison between few-shot prompting and *super neurons* on LLaVA-v1.5-7b.

| Dataset | Shots | Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|---|
| POPE | 0 | 88.7 | 88.7 | 89.6 | 89.1 |
| | 1 | 90.1(+1.4) | **94.7**(+6.0) | 85.7(-3.9) | 90.0(+0.9) |
| | 3 | 88.4(-0.3) | 94.3(+5.6) | 82.5(-7.1) | 88.0(-1.1) |
| | 5 | 88.4(-0.3) | 94.1(+5.4) | 82.5(-7.1) | 88.0(-1.1) |
| | SNs | **93.2**(+4.5) | 93.7(+5.0) | **93.1**(+3.5) | **93.4**(+4.3) |
| INSTAORDER (DEPTH) | 0 | 60.2 | 61.6 | **82.6** | 70.6 |
| | 1 | 57.9(-2.3) | 66.6(+5.0) | 54.8(-27.8) | 60.2(-10.4) |
| | 3 | 61.5(+1.3) | 63.5(+1.9) | 79.0(-3.6) | 70.4(-0.2) |
| | 5 | 61.5(+1.3) | 63.9(+2.3) | 77.1(-5.5) | 69.9(-0.7) |
| | SNs | **68.9**(+8.7) | **74.3**(+12.7) | 70.7(-11.9) | **72.5**(+1.9) |
| INSTAORDER (OCC.) | 0 | 24.6 | 13.5 | 75.8 | 22.9 |
| | 1 | **84.2**(+59.6) | 13.0(-0.5) | 1.2(-74.6) | 2.1(-20.8) |
| | 3 | 25.9(+1.3) | 14.7(+1.2) | **83.0**(+7.2) | 24.9(+2.0) |
| | 5 | 38.2(+13.6) | 15.0(+1.5) | 68.1(-7.7) | 24.6(+1.7) |
| | SNs | 67.1(+42.5) | **23.8**(+10.3) | 55.4(-20.4) | **33.3**(+10.4) |
| VIZWIZ | 0 | 63.4 | 58.6 | 91.4 | 71.4 |
| | 1 | 64.5(+1.1) | 59.6(+1.0) | 90.3(-1.1) | 71.8(+0.4) |
| | 3 | 64.0(+0.6) | 59.3(+0.7) | 89.2(-2.2) | 71.2(-0.2) |
| | 5 | 66.1(+2.7) | 60.6(+2.0) | **92.5**(+1.1) | **73.2**(+1.8) |
| | SNs | **72.6**(+9.2) | **73.9**(+15.3) | 69.9(-21.5) | 71.8(+0.4) |
| CLEVR | 0 | 51.1 | 50.7 | **79.1** | **61.8** |
| | 1 | 52.1(+1.0) | 51.5(+0.8) | 71.7(-7.4) | 60.0(-1.8) |
| | 3 | 52.3(+1.2) | 51.6(+0.9) | 72.8(-6.3) | 60.4(-1.4) |
| | 5 | 51.7(+0.6) | 51.2(+0.5) | 72.7(-6.4) | 60.1(-1.7) |
| | SNs | **53.5**(+2.4) | **54.0**(+3.3) | 47.7(-31.4) | 50.6(-11.2) |
| A-OKVQA | 0 | 54.7 | 34.9 | **93.4** | 50.8 |
| | 1 | 73.8(+19.1) | 48.5(+13.6) | 81.7(-11.7) | 60.9(+10.1) |
| | 3 | **74.4**(+19.7) | **49.2**(+14.3) | 83.1(-10.3) | **61.9**(+11.1) |
| | 5 | 74.3(+19.6) | **49.2**(+14.3) | 83.6(-9.8) | **61.9**(+11.1) |
| | SNs | 74.0(+19.3) | 48.7(+13.8) | 80.0(-13.4) | 60.6(+9.8) |
| SCIENCEQA | 0 | 53.5 | 40.1 | **85.9** | 54.7 |
| | 1 | 64.9(+11.4) | 47.4(+7.3) | 68.1(-17.8) | 56.0(+1.3) |
| | 3 | 68.1(+14.6) | 51.0(+10.9) | 64.5(-21.4) | 57.0(+2.3) |
| | 5 | **69.0**(+15.5) | **52.2**(+12.1) | 61.1(-24.8) | 56.3(+1.6) |
| | SNs | 64.9(+11.4) | 47.7(+7.6) | 76.6(-9.3) | **58.8**(+4.1) |

Here, $\mathbb{1}$ is the indicator function.

### C.2 MAIN BENCHMARK DISCUSSIONS

**Summary.** *Super neurons* obtain strong performance on all benchmarks, as testified by fig. 2, in the main manuscript and tables 12 and 13 of the supplementary materials.

**Detailed results of *super neurons* on LLaVA-v1.5-7b results.** Specifically, LLaVA-v1.5-7b's *super neurons* outperform the base model on all datasets. Detailed results show that the F1 of *super neurons* is above the F1 of the base model on all datasets, except on CLEVR (table 12). We suspect that this might be due to the fact that LLaVA-v1.5-7b has been trained on visual synthetic data in both its pretrained and alignment stages. This seems confirmed by the fact that we can still find highly performant *super neurons* on other spatial understanding datasets such as INSTAORDER (OCC.) and INSTAORDER (DEPTH) (table 12).

Table 11: Comparison between few-shot prompting and *super neurons* on Qwen3-VL-8b-Instruct.

| Dataset | Shots | Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|---|
| POPE | 0 | 92.0 | 92.0 | 92.0 | 92.0 |
| | 1 | 92.8(+0.8) | 97.7(+5.7) | 88.1(-3.9) | 92.6(+0.6) |
| | 3 | 92.2(+0.2) | 98.0(+6.0) | 86.7(-5.3) | 92.0 |
| | 5 | 91.8(-0.2) | **98.2**(+6.2) | 85.5(-6.5) | 91.4(-0.6) |
| | SNs | **93.8**(+1.8) | **95.8**(+3.8) | **92.1**(+0.1) | **93.9**(+1.9) |
| INSTAORDER (DEPTH) | 0 | 46.9 | **84.3** | 10.1 | 18.1 |
| | 1 | 55.4(+8.5) | 77.3(-7.0) | 32.4(+22.3) | 45.7(+27.6) |
| | 3 | 59.2(+12.3) | 77.9(-6.4) | 41.2(+31.1) | 53.9(+35.8) |
| | 5 | 59.7(+12.8) | 77.3(-7.0) | 43.1(+33.0) | 55.4(+37.3) |
| | SNs | **67.6**(+20.7) | 76.1(-8.2) | **64.1**(+54.0) | **69.6**(+51.5) |
| INSTAORDER (OCC.) | 0 | 84.6 | 20.3 | 1.5 | 2.8 |
| | 1 | **85.2**(+0.6) | 13.6(-6.7) | 0.0(-1.5) | 0.0(-2.8) |
| | 3 | 14.9(-69.7) | 14.8(-5.5) | **99.5**(+98.0) | 25.7(+22.9) |
| | 5 | 23.7(-60.9) | 14.8(-5.5) | 87.1(+85.6) | 25.3(+22.5) |
| | SNs | 68.4(-16.2) | **24.5**(+4.2) | 54.3(+52.8) | **33.7**(+30.9) |
| VIZWIZ | 0 | 77.4 | 72.2 | 89.2 | 79.8 |
| | 1 | 76.3(-1.1) | 74.3(+2.1) | 80.6(-8.6) | 77.3(-2.5) |
| | 3 | 79.6(+2.2) | 74.8(+2.6) | 89.2 | 81.4(+1.6) |
| | 5 | 78.5(+1.1) | 73.0(+0.8) | **90.3**(+1.1) | 80.8(+1.0) |
| | SNs | **81.7**(+4.3) | **80.4**(+8.2) | 83.9(-5.3) | **82.1**(+2.3) |
| CLEVR | 0 | 65.8 | 59.4 | **99.5** | 74.4 |
| | 1 | 80.9(+15.1) | 73.3(+13.9) | 97.1(-2.4) | 83.5(+9.1) |
| | 3 | 75.9(+10.1) | 68.3(+8.9) | 96.6(-2.9) | 80.0(+5.6) |
| | 5 | 75.6(+9.8) | 68.7(+9.3) | 94.1(-5.4) | 79.4(+5.0) |
| | SNs | **93.5**(+27.7) | **94.5**(+35.1) | 92.2(-7.3) | **93.4**(+19.0) |
| A-OKVQA | 0 | 83.6 | 62.2 | 87.7 | 72.8 |
| | 1 | 83.2(-0.4) | 61.3(-0.9) | **89.0**(+1.3) | 72.6(-0.2) |
| | 3 | 83.2(-0.4) | 61.4(-0.8) | 88.6(+0.9) | 72.5(-0.3) |
| | 5 | 84.0(+0.4) | 62.6(+0.4) | 88.7(+1.0) | 73.4(+0.6) |
| | SNs | **86.9**(+3.3) | **69.5**(+7.3) | 84.5(-3.2) | **76.3**(+3.5) |
| SCIENCEQA | 0 | 82.3 | **69.8** | 80.8 | 74.9 |
| | 1 | 79.5(-2.8) | 64.1(-5.7) | 84.7(+3.9) | 73.0(-1.9) |
| | 3 | 80.0(-2.3) | 64.7(-5.1) | 85.6(+4.8) | 73.7(-1.2) |
| | 5 | 80.4(-1.9) | 65.4(-4.4) | 84.8(+4.0) | 73.8(-1.1) |
| | SNs | **83.5**(+1.2) | **69.8** | **87.1**(+6.3) | **77.5**(+2.6) |

Nonetheless, *super neurons* consistently improve the performance of the base model with substantial gains on the saturated benchmark of POPE (+4.3 F1 points) or MCQ datasets like A-OKVQA (+9.8 F1 points) and SCIENCEQA (+4.1 F1 points) (table 12). On average, $k = 10$ *super neurons* increase the F1 score of LLaVA-v1.5-7b by 2.8 points (table 15a).

**Detailed results of *super neurons* on Qwen3-VL-8b-Instruct.** We also report strong results for Qwen3-VL-8b-Instruct's *super neurons*. The accuracy of its *super neurons* outperforms the model on all datasets, except on INSTAORDER (OCC.) (fig. 2). As specified in section 4 of the main manuscript, we found that the base model exploits dataset statistics, since the percentage of "no" labels is extremely close to its performance. This indicates that the model simply answers "no" to all questions without genuinely trying to propose an accurate answer. Looking at the detailed performance in table 13, we report that Qwen3-VL-8b-Instruct has indeed a blind spot for instance-wise occlusion understanding, as its F1 is 2.8, even lower than LLaVA-v1.5-7b, although it is a more recent and better network. We are confident this is not a bug from our implementation since we prompt the model the same way for all datasets and Qwen3-VL-8b-Instruct and its *super neurons* obtain generally strong results on them. Moreover, it is performing well on INSTAORDER (DEPTH),

Table 12: Full comparison between sparse attention vectors and *super neurons* on LLaVA-v1.5-7b.

| Dataset | Metric | LLaVA-v1.5-7b | SAVs | SNs |
|---|---|---|---|---|
| POPE | Accuracy | 88.7 | 91.1(+2.4) | **93.2**(+4.5) |
| | Precision | 88.7 | **95.0**(+6.3) | 93.7(+5.0) |
| | Recall | 89.6 | 87.4(-2.2) | **93.1**(+3.5) |
| | F1 | 89.1 | 91.0(+1.9) | **93.4**(+4.3) |
| INSTAORDER (DEPTH) | Accuracy | 60.2 | 51.7(-8.5) | **68.9**(+8.7) |
| | Precision | 61.6 | 60.5(-1.2) | **74.3**(+12.7) |
| | Recall | **82.6** | 48.0(-34.6) | 70.7(-11.9) |
| | F1 | 70.6 | 53.5(-17.1) | **72.5**(+1.9) |
| INSTAORDER (OCC.) | Accuracy | 24.6 | 47.9(+23.3) | **67.1**(+42.5) |
| | Precision | 13.5 | 17.9(+4.4) | **23.8**(+10.3) |
| | Recall | **75.8** | 70.3(-5.4) | 55.4(-20.4) |
| | F1 | 22.9 | 28.5(+5.6) | **33.3**(+10.4) |
| VIZWIZ | Accuracy | 63.4 | 64.5(+1.1) | **72.6**(+9.2) |
| | Precision | 58.6 | 72.9(+14.3) | **73.9**(+15.3) |
| | Recall | **91.4** | 46.2(-45.2) | 69.9(-21.5) |
| | F1 | 71.4 | 56.6(-14.9) | **71.8**(+0.4) |
| CLEVR | Accuracy | 51.1 | 49.5(-1.6) | **53.5**(+2.4) |
| | Precision | 50.7 | 49.6(-1.1) | **54.0**(+3.3) |
| | Recall | **79.1** | 61.3(-17.8) | 47.7(-31.4) |
| | F1 | **61.8** | 54.8(-7.0) | 50.6(-11.2) |
| A-OKVQA | Accuracy | 54.7 | 71.8(+17.1) | **74.0**(+19.3) |
| | Precision | 34.9 | 46.3(+11.4) | **48.7**(+13.8) |
| | Recall | **93.4** | 78.3(-15.0) | 80.0(-13.4) |
| | F1 | 50.8 | 58.2(+7.4) | **60.6**(+9.8) |
| SCIENCEQA | Accuracy | 53.5 | 63.9(+10.4) | **64.9**(+11.4) |
| | Precision | 40.1 | 46.3(+6.2) | **47.7**(+7.6) |
| | Recall | **85.9** | 65.6(-20.3) | 76.6(-9.3) |
| | F1 | 54.7 | 54.3(-0.4) | **58.8**(+4.1) |

which uses the same prompt format (table 13). Despite this, *super neurons* increase the performance of the model by a remarkable 30.9 points, indicating the model still possesses internal representations able to at least comprehend the task (table 13).

Generally speaking, *super neurons* of Qwen3-VL-8b-Instruct are particularly good at spatial understanding while still obtaining strong results on datasets requiring common knowledge like A-OKVQA (+3.5 F1 points) and SCIENCEQA (+2.6 F1 points) (table 13). Overall, $k = 10$ *super neurons* of Qwen3-VL-8b-Instruct improve the F1 performance of its base model by a substantial 16.0 F1 points (table 15b).

### C.3 COMPARISON WITH FEW-SHOT PROMPTING

**Summary.** Tables 10 and 11 indicate that *super neurons* are overall more performant than few-shot prompting.

**Results.** In some cases, few-shot learning obtains lower performance than the base model (LLaVA-v1.5-7b, on POPE, on 3 and 5 shots, or INSTAORDER (DEPTH) on all shots). While unexpected, this aligns with previous findings on few-shot prompting (Mitra et al., 2025). We see diminishing returns of F1 for few-shot prompting on most datasets when the number of shots grows above 3. For LLaVA-v1.5-7b, we report that 5-shot prompting beats $k = 10$ *super neurons* on SCIENCEQA accuracy, however, F1 obtained using that strategy is much lower than F1 obtained by *super neurons*. On VIZWIZ, our approach shows lower F1 but increasing $k$ from 10 to 20 uplifts the F1 to 75.4, comfortably outperforming few-shot prompting.

Table 13: Comparison between SAVs and *super neurons* on Qwen3-VL-8b-Instruct.

| Dataset | Metric | Qwen3-VL-8b-Instruct | SAVs | SNs |
|---|---|---|---|---|
| POPE | Accuracy | 92.0 | 93.0(+1.0) | **93.8**(+1.8) |
| | Precision | **98.0** | 97.8(-0.2) | 95.8(-2.2) |
| | Recall | 86.2 | 88.3(+2.1) | **92.1**(+5.9) |
| | F1 | 91.7 | 92.8(+1.1) | **93.9**(+2.2) |
| INSTAORDER (DEPTH) | Accuracy | 46.9 | 56.5(+9.7) | **67.6**(+20.7) |
| | Precision | **84.3** | 64.6(-19.7) | 76.1(-8.2) |
| | Recall | 10.1 | 55.1(+45.0) | **64.1**(+54.0) |
| | F1 | 18.1 | 59.5(+41.4) | **69.6**(+51.5) |
| INSTAORDER (OCC.) | Accuracy | **84.6** | 56.9(-27.7) | 68.4(-16.2) |
| | Precision | 20.3 | 17.9(-2.4) | **24.5**(+4.2) |
| | Recall | 1.5 | 53.3(+51.8) | **54.3**(+52.8) |
| | F1 | 2.8 | 26.8(+24.0) | **33.7**(+30.9) |
| VIZWIZ | Accuracy | 77.4 | 66.7(-10.8) | **81.7**(+4.3) |
| | Precision | 72.2 | 70.1(-2.0) | **80.4**(+8.2) |
| | Recall | **89.2** | 58.1(-31.2) | 83.9(-5.3) |
| | F1 | 79.8 | 63.5(-16.3) | **82.1**(+2.3) |
| CLEVR | Accuracy | 65.8 | 92.8(+27.0) | **93.5**(+27.7) |
| | Precision | 59.4 | 89.9(+30.5) | **94.5**(+35.1) |
| | Recall | **99.5** | 96.6(-2.9) | 92.2(-7.3) |
| | F1 | 74.4 | 93.1(+18.7) | **93.4**(+19.0) |
| A-OKVQA | Accuracy | 83.6 | 82.9(-0.7) | **86.9**(+3.3) |
| | Precision | 62.2 | 60.8(-1.4) | **69.5**(+7.3) |
| | Recall | 87.7 | **88.5**(+0.8) | 84.5(-3.2) |
| | F1 | 72.8 | 72.1(-0.7) | **76.3**(+3.5) |
| SCIENCEQA | Accuracy | 82.3 | 77.9(-4.4) | **83.5**(+1.2) |
| | Precision | **69.8** | 61.5(-8.3) | 69.8(+0.0) |
| | Recall | 80.8 | 86.5(+5.7) | **87.1**(+6.3) |
| | F1 | 74.9 | 71.9(-3.0) | **77.5**(+2.6) |

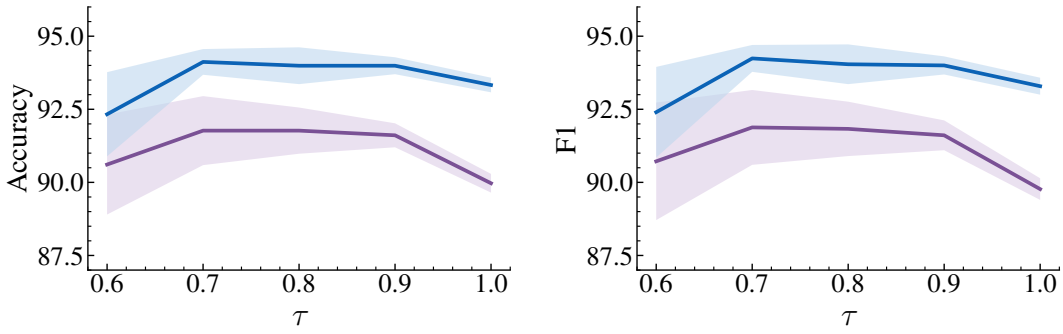
**Trade-offs.** Few-shot prompting can be useful to improve the performance of the model, but that also comes at the cost of increased memory and at the cost of diminishing context length during generation. On the other hand, *super neurons* do not require additional memory at inference, nor sacrifice context length since they do not generate any tokens.

### C.4 COMPARISON WITH SAVS

**Summary.** We find that *super neurons* outperform SAVs on almost all benchmarks, for both LLaVA-v1.5-7b and Qwen3-VL-8b-Instruct (tables 12 and 13).

**Results with respect to the design.** Since SAVs are selected based on the accuracy of their attention heads, they mainly improve the validation accuracy, while sometimes degrading F1. For example, SAVs extracted from Qwen3-VL-8b-Instruct on VIZWIZ the F1 by 16.3 points (table 13). On the other hand, since our *super neurons* are discovered from the BCE with respect to the ground truth, they encode more robust signals and improve both accuracy and F1 more reliably (tables 12 and 13).

**Results.** While SAVs outperform the model on common benchmarks such as A-OKVQA or SCIENCEQA and score especially well on POPE none of their results is able to outperform *super neurons*. This makes sense since SAVs were initially designed with the intent of improving performance on object-centric datasets such as POPE. When questions become more complex and abstract, such as the relation between objects on SCIENCEQA, performance drop (tables 12 and 13). While SAVs




(a) Accuracy of *super neurons* for different $\tau$.

(b) F1 of *super neurons* for different $\tau$.

Figure 9: Ablation experiment on $\tau$. We sweep its value from 0.6 to 0.9 and report the resulting accuracy and F1 of both LLaVA-v1.5-7b and Qwen3-VL-8b-Instruct's *super neurons* on POPE. We report that $\tau = 0.7$ provides the best results for both metrics, on both models.

obtain a higher F1 than *super neurons* on LLaVA-v1.5-7b on CLEVR, none of the approach is able to outperform the base model.

As discussed in section 4 of the main manuscript, we believe that the extended search space for efficient parameters offered by our *super neurons*' might result in increasing the diversity of the signals, and therefore allowing to outperform the model more comfortably.

### C.5 DECORRELATION ANALYSIS

**Intuition.** The greedy decorrelation routine in our algorithm (algorithm 1 of the main manuscript) is a key component for selecting a highly performant ensembles of *super neurons*. Intuitively, a higher $\tau$ lets *super neurons* carrying similar signals be aggregated in the final ensemble, as long as they are top performing units. Conversely, a lower $\tau$ rejects adding a particular unit to the ensemble, should its signal be correlated with a unit previously added in the ensemble. In that case, while the algorithm would look for lower performing units, these *super neurons* would still positively contribute to the performance of the ensemble thanks to their uncorrelated signal contributions.

**Results.** In fig. 9, we sweep the decorrelation hyperparameter $\tau$ from 0.6 to 1.0 on POPE to understand its impact on the resulting performance of the selected *super neurons*. While all values between 0.7 and 0.9 are valid choices for maximizing the performance of the final *super neurons* ensemble, $\tau = 0.7$ obtains the best accuracy and F1 (figs. 9a and 9b of the supplementary materials). Thus, we default to this value for all our experiments.

When $\tau = 1.0$, *i.e.*, when we let the best units form the final ensemble, we observe substantial performance degradation. This setting is essentially equivalent to completely removing the decorrelation routine. This indicates that a level of diversity in the signal contribution is required in order to maximize the performance of *super neurons*. On the other hand, a strict decorrelation policy, such as $\tau = 0.6$ also induces lower performance (fig. 9). This is most likely because highly decorrelated *super neurons* are not available in the most performant units, thus the algorithm needs to look at lower performing units, which, in turn, hurts the overall performance (fig. 9).

### C.6 SCALING *Super neurons*

The Qwen-VL-Instruct family of model is dense models ranging from 2b to 32b parameters. We report the scaling properties of this family in fig. 5 of the main manuscript, and table 14. In these experiments, we set $k = 10$ and observe constant improvement in the performance of *super neurons*, to the extent that their relative score against their base model actually increases too.

When scaling to 32b parameters, we see a slight drop of absolute performance with respect to the 8b model although the base 32b model performs better than the base 8b model. We believe that this is due to the fact that models from 2b to 8b were distilled from the 32b model after extensive pretraining (Qwen Team, 2025). In this setting, the performance of the 32b model might be condensed

Table 14: Scaling the Qwen-VL-Instruct family of models on POPE.

| Model | Method | Acc. | Prec. | Recall | F1 |
|---|---|---|---|---|---|
| Qwen3-VL-2b-Instruct | Model | 91.8 | **97.7** | 86.1 | 91.6 |
| | SNs | **92.7**(+0.9) | 95.9(-1.8) | **89.7**(+3.6) | **92.7**(+1.1) |
| Qwen3-VL-4b-Instruct | Model | 91.5 | **98.5** | 84.9 | 91.2 |
| | SNs | **93.1**(+1.6) | 97.1(-1.4) | **89.3**(+4.4) | **93.1**(+1.9) |
| Qwen3-VL-8b-Instruct | Model | 92.0 | **98.0** | 86.2 | 91.7 |
| | SNs | **93.8**(+1.8) | 95.8(-2.2) | **92.1**(+5.9) | **93.9**(+2.2) |
| Qwen3-VL-32b-Instruct | Model | 92.7 | **97.8** | 87.9 | 92.6 |
| | SNs | **93.6**(+0.9) | 96.3(-1.5) | **91.1**(+3.2) | **93.6**(+1.0) |

Table 15: Macro-average performance of *super neurons* across all seven datasets as the number of selected neurons $k$ increases.

(a) LLaVA-v1.5-7b.

| Method | Avg. Acc. | Avg. F1 |
|---|---|---|
| Model | 56.6 | 60.2 |
| $k = 1$ | 66.6(+10.0) | 59.6(-0.6) |
| $k = 2$ | 68.9(+12.3) | 61.5(+1.3) |
| $k = 5$ | 70.6(+14.0) | 63.2(+3.0) |
| $k = 10$ | 70.6(+14.0) | 63.0(+2.8) |
| $k = 20$ | **71.5**(+14.9) | **64.0**(+3.8) |
| $k = 50$ | 71.0(+14.4) | 63.9(+3.7) |
| $k = 100$ | 71.2(+14.6) | **64.0**(+3.8) |

(b) Qwen3-VL-8b-Instruct.

| Method | Avg. Acc. | Avg. F1 |
|---|---|---|
| Model | 76.1 | 59.2 |
| $k = 1$ | 81.1(+5.0) | 73.3(+14.1) |
| $k = 2$ | 80.8(+4.7) | 74.0(+14.8) |
| $k = 5$ | 81.6(+5.5) | 74.6(+15.4) |
| $k = 10$ | **83.2**(+7.1) | 75.2(+16.0) |
| $k = 20$ | 82.7(+6.6) | 75.8(+16.6) |
| $k = 50$ | 82.7(+6.6) | **76.0**(+16.7) |
| $k = 100$ | 82.7(+6.6) | 75.9(+16.7) |

in particular areas of the student models, explaining why few units might obtain better performance. While hypothetical, this claim is supported by the observation that increasing the number of units of Qwen3-VL-8b-Instruct from 10 to 100 improves *super neurons*' performance to 94.2 accuracy and 94.4 F1. This indicates that performance continues to improve as we scale the number of selected units.

### C.7 ANALYSIS OF THE NUMBER OF *Super neurons*

**Number of *Super neurons*.** In table 15, we show the impact of increasing the number of *Super neurons* for inference for both LLaVA-v1.5-7b (table 15a), Qwen3-VL-8b-Instruct (table 15b). Specifically, we compute the macro-average of the accuracy and F1 on all the benchmarks of our dataset suite.

Results indicate that a single *super neuron* is enough to outperform the accuracy of the models. However, we also observe that different models achieve maximum performance at different regimes of $k$. As a matter of fact, LLaVA-v1.5-7b reaches its average top performance when $k = 20$, while Qwen3-VL-8b-Instruct attains it when $k = 50$. Generally, increasing the number of units of the ensemble of *super neurons* increases performance (fig. 7).

There is no trade-off in increasing $k$ since our algorithm ranks all units in the first place in an offline fashion. At inference, we just keep slightly more neural indices.

In our manuscript, we set the value of $k$ to 10 since it is (1) one of the strongest options (2) the first two-digit value (3) a memorable and quickly recognizable value. Even when we do not set $k$ to its best value per model, *super neurons* consistently outperform the baselines in the main benchmarks.

Table 16: Detailed results of LLaVA-v1.5-7b on MM-VET after being finetuned on INSTAORDER (OCC.). Scores are obtained using the official online evaluator, with GPT-4.1 as a judge.

| Metric | Base | LoRA | SFT | SNs |
|---|---|---|---|---|
| gen_know_rec | 15.5 | 0.8(-14.7) | 0.2(-15.3) | **18.5**(+3.0) |
| rec | **65.7** | 16.2(-49.5) | 21.9(-43.8) | 59.2(-6.5) |
| ocr_spat | 16.2 | 0.0(-16.2) | 0.0(-16.2) | **24.6**(+8.4) |
| ocr_math_spat | 14.3 | 0.7(-13.6) | 0.0(-14.3) | **22.1**(+7.8) |
| rec_spat | 58.3 | 16.7(-41.6) | 17.5(-40.8) | **75.0**(+16.7) |
| ocr | **52.5** | 0.8(-51.7) | 0.0(-52.5) | 43.3(-9.2) |
| ocr_math | 0.0 | 0.0 | 0.0 | **0.9**(+0.9) |
| know_rec | 16.7 | 2.2(-14.5) | 2.2(-14.5) | **28.9**(+12.2) |
| ocr_gen_rec_spat | **31.2** | 0.0(-31.2) | 0.0(-31.2) | 25.0(-6.2) |
| ocr_gen_know_rec | **22.5** | 1.2(-21.3) | 1.2(-21.3) | 16.2(-6.3) |
| ocr_rec_spat | 14.3 | 0.0(-14.3) | **28.6**(+14.3) | 14.3 |
| ocr_rec | **50.0** | 25.0(-25.0) | 25.0(-25.0) | **50.0** |
| ocr_know_spat | **0.0** | **0.0** | **0.0** | **0.0** |
| know_rec_spat | **0.0** | **0.0** | **0.0** | **0.0** |
| ocr_gen_spat | **25.0** | 0.0(-25.0) | 0.0(-25.0) | 10.0(-15.0) |
| ocr_rec_math_spat | **0.0** | **0.0** | **0.0** | **0.0** |
| **Total** | 28.8 | 4.6(-24.2) | 6.2(-22.6) | **30.4**(+1.6) |

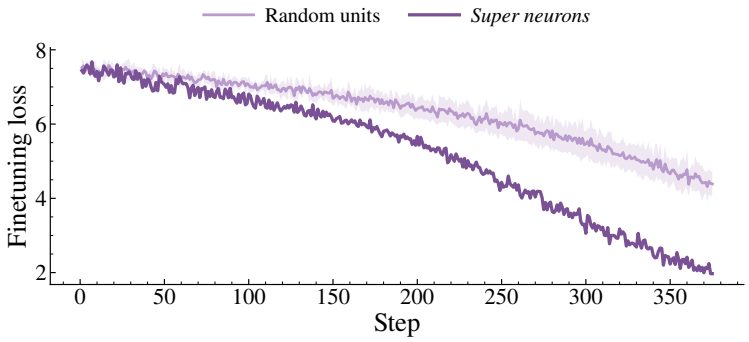



Figure 10: Finetuning dynamics of $k = 10$ best *super neurons* and 10 randomly selected units within LLaVA-v1.5-7b. We use $N = 3{,}000$ samples from the INSTAORDER (OCC.) train set. The shaded area represents the 95% confidence interval, computed across 5 different runs.

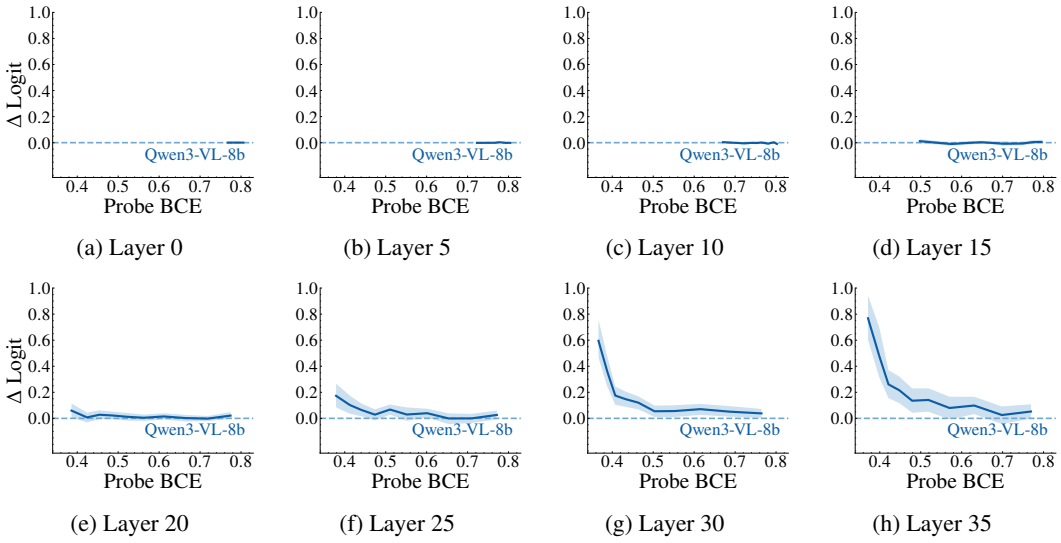
### C.8 FINETUNING RESULTS

**MM-VET benchmark.** In table 16, we report the detailed results obtained by LLaVA-v1.5-7b on MM-VET (Yu et al., 2024) after being finetuned on INSTAORDER (OCC.) using the protocol detailed in appendix section A.4.

As hypothesized, *super neurons* finetuning maintains the general capabilities of the model (table 16). What is more surprising is that finetuning the model on INSTAORDER (OCC.) *increases* the overall performance of the model on MM-VET (table 16). We observe that the categories related to spatial understanding (spat) almost all improve after being finetuned on instance-wise occlusion prediction, clearly demonstrating the task transfer success.

**Training dynamics.** Using the same protocol, we compare the training dynamics between the 10 best *Super neurons* and 10 randomly selected neurons in the model. To account for randomness, we run the randomly sampled neurons experiment 5 times and plot the 95% confidence interval of these runs. We show the finetuning loss of both these runs in fig. 10.

While the dynamics initially match, *super neurons* finetuning starts to better optimize towards the target task around step 100. As finetuning progresses, the gap between the two set of neurons continues to widen. At the end of the training, for that annotation budget, *super neurons* converge




(a) Layer 0 (b) Layer 5 (c) Layer 10 (d) Layer 15

(e) Layer 20 (f) Layer 25 (g) Layer 30 (h) Layer 35

Figure 11: Causal effect of patching single units of Qwen3-VL-8b-Instruct on POPE with respect to their probe BCE, for layers 0 to 35, in steps of 5. Per layer, 1,000 units evenly spaced along the layer's BCE ranking are grouped into 10 equal-count bins. The solid line is the bin mean plotted at the bin's BCE midpoint, and the shaded area represents the bootstrap 95% confidence interval of that mean.

much better than optimizing 10 randomly selected units. This emphasizes that the selected *super neurons* on INSTAORDER (OCC.) are learning the instance-wise occlusion prediction task much better than randomly selected units, and would thus be indeed task experts.

**Connection with the causality analysis.** The fact that *super neurons* optimize for the instance-wise occlusion prediction task much better than randomly selected units (fig. 10) can be directly linked with our findings on the causal roles of *super neurons* (fig. 4, in the main manuscript, and fig. 11 in appendix D). Since *super neurons* play a greater role in logit variation than other units in the model, it makes sense to observe that they can be optimized for the target task much better than randomly selected units, which have less contribution to the generated answer from the VLM.

## D ADDITIONAL CAUSAL EXPERIMENTS

**Implementation.** We patch the activations of specific neurons in the output hidden state of the Qwen3-VL-8b-Instruct LLM at the last prompt token, which is where the *super neurons* are selected from. For each layer, we rank neurons by their probing BCE, and select 100 units from each decile bin with respect to their ranks. Each unit's activation is patched to $z = 25$ in the direction the probe predicts to move the answer away from the ground truth. Then, we report the resulting change in the yes/no logit margin of the first generated token, averaged over $N = 500$ POPE questions.

**Results.** Figure 11 shows that informativeness precedes causal influence. The most informative neuron's BCE reaches 0.5 in layer 15 (fig. 11d) and plateaus at 0.36 at layer 20 (fig. 11e), while the causally effective neurons start to appear around layer 25 (fig. 11f). Moreover, within the same layer, more informative neurons, with lower BCE, lead to greater logit margin. Since our algorithm selects low BCE neurons as *super neurons*, they are also expected to have greater causal influence. It is also worth noting that most *super neurons* for POPE are selected after layer 25, which is where the causally effective neurons start to appear.

# E QUALITATIVE RESULTS

**Confidence score.** The confidence of the model, $c_{f_\theta}$, is computed from the logits of the first generated token. We compute the probability of yes over no using the difference of the logits $o_{\text{yes}}, o_{\text{no}}$. The confidence of the *super neurons*, $c_{\text{SN}}$, is computed using the activations from the last prompt token. We treat each signed activation as an independent logit and average them over $\mathcal{S}^\star$, and apply a sigmoid:

$$c_{f_\theta} = \text{sigmoid}(o_{\text{yes}} - o_{\text{no}}), \qquad c_{\text{SN}} = \text{sigmoid}\left(\frac{1}{k}\sum_{(l,d)\in\mathcal{S}^\star} s_{l,d}\, z_{l,d}\right). \tag{7}$$

**Results.** We highlight a few more qualitative results obtained with *super neurons* in fig. 12. *Super neurons* are accurate on synthetic spatial understanding data, as shown in the CLEVR example (fig. 12a), but also on real-world spatial understanding, even when target objects are small and precisely defined (figs. 12b and 12c). Their commonsense reasoning (figs. 12b and 12e) and precise understanding of ambiguous images (fig. 12d) are also noteworthy.

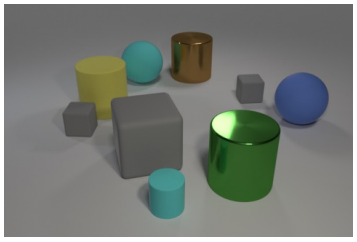
Most importantly, since the multiple choice questions are cast as a series of binary questions, we observe an interesting tendency for the models to answer positively to many answers when unsure of the true answer (figs. 12b, 12e and 12f). This is not the case of *super neurons*, which always answer sharply with a single right answer.

***Super neurons* cartography.** We plot the locations of the best $k = 10$ *super neurons* in their respective models for each dataset in figs. 13 and 14.

We observe a similar trend between the two models, where most of the *super neurons* are usually located towards the middle or the deepest layers of the models. This is especially the case on CLEVR from LLaVA-v1.5-7b (fig. 13e) and INSTAORDER (DEPTH), from Qwen3-VL-8b-Instruct's respective *super neurons* (fig. 14b).

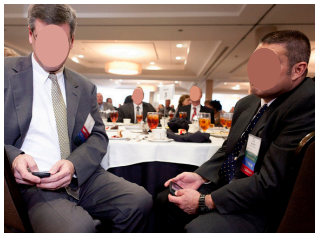
This aligns with the causality observation developed in section D and particularly in fig. 11, where we observe that lower BCE units, *i.e.*, *super neurons*, of Qwen3-VL-8b-Instruct appear towards layer 20 and 25, and up to the final layers of the model.

Nonetheless, strong *super neurons* seem to appear significantly earlier on POPE and INSTAORDER (OCC.) on LLaVA-v1.5-7b (figs. 13a and 13c).



Is there anything else that is the same color as the large rubber cylinder?
Ground truth: **no**
Baseline: yes (confidence: 97.0%)
SNs: no (confidence: 66.0%)

(a) Qwen3-VL-8b-Instruct on CLEVR.



What is the man in the gray suit on the left looking down to check?
Choices:
1. A phone
2. A tablet
3. A notebook
4. A page

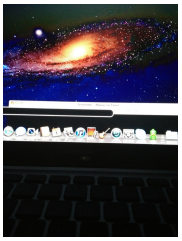
Ground truth: **a phone**
Baseline: a phone; a tablet; a pager
SNs: a phone

(b) LLaVA-v1.5-7b on A-OKVQA.



Is the elephant at position [426, 143, 560, 327] in front of the elephant at position [52, 193, 85, 222] from the viewer's perspective?
Ground truth: **yes**
Baseline: no (confidence: 67.0%)
SNs: yes (confidence: 88.0%)

(c) Qwen3-VL-8b-Instruct on INSTAORDER (DEPTH).



Could you tell me if there's a captcha? And if there is, what does it say for the website?
Ground truth: **no**
Baseline: yes (confidence: 98.0%)
SNs: no (confidence: 70.0%)

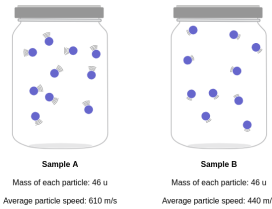
(d) LLaVA-v1.5-7b on VIZWIZ.



This fork has how many prongs?
Choices:
1. Six
2. Seven
3. Five
4. Three

Ground truth: **Three**
Baseline: six; seven; five; three
SNs: three

(e) Qwen3-VL-8b-Instruct on A-OKVQA.

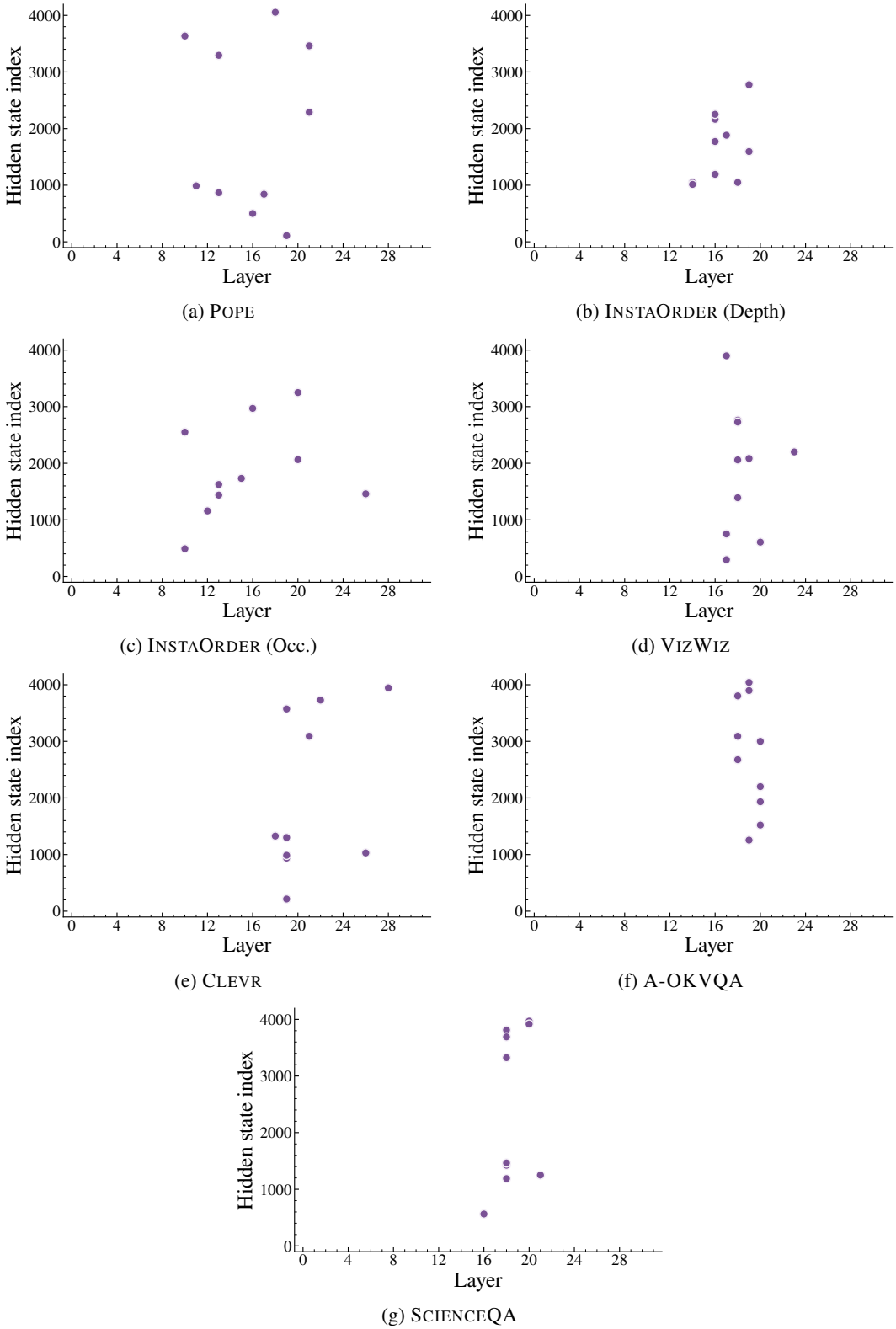



Compare the average kinetic energy of the particles in each sample. Which sample has the highest temperature?
Choices:
1. B
2. Neither
3. A

Ground truth: **A**
Baseline: B; Neither; A
SNs: A

(f) LLaVA-v1.5-7b on SCIENCEQA.

Figure 12: Qualitative results of *super neurons* on both LLaVA-v1.5-7b and Qwen3-VL-8b-Instruct. We also show the model prediction in each case. We write the confidence of each method next to each prediction.



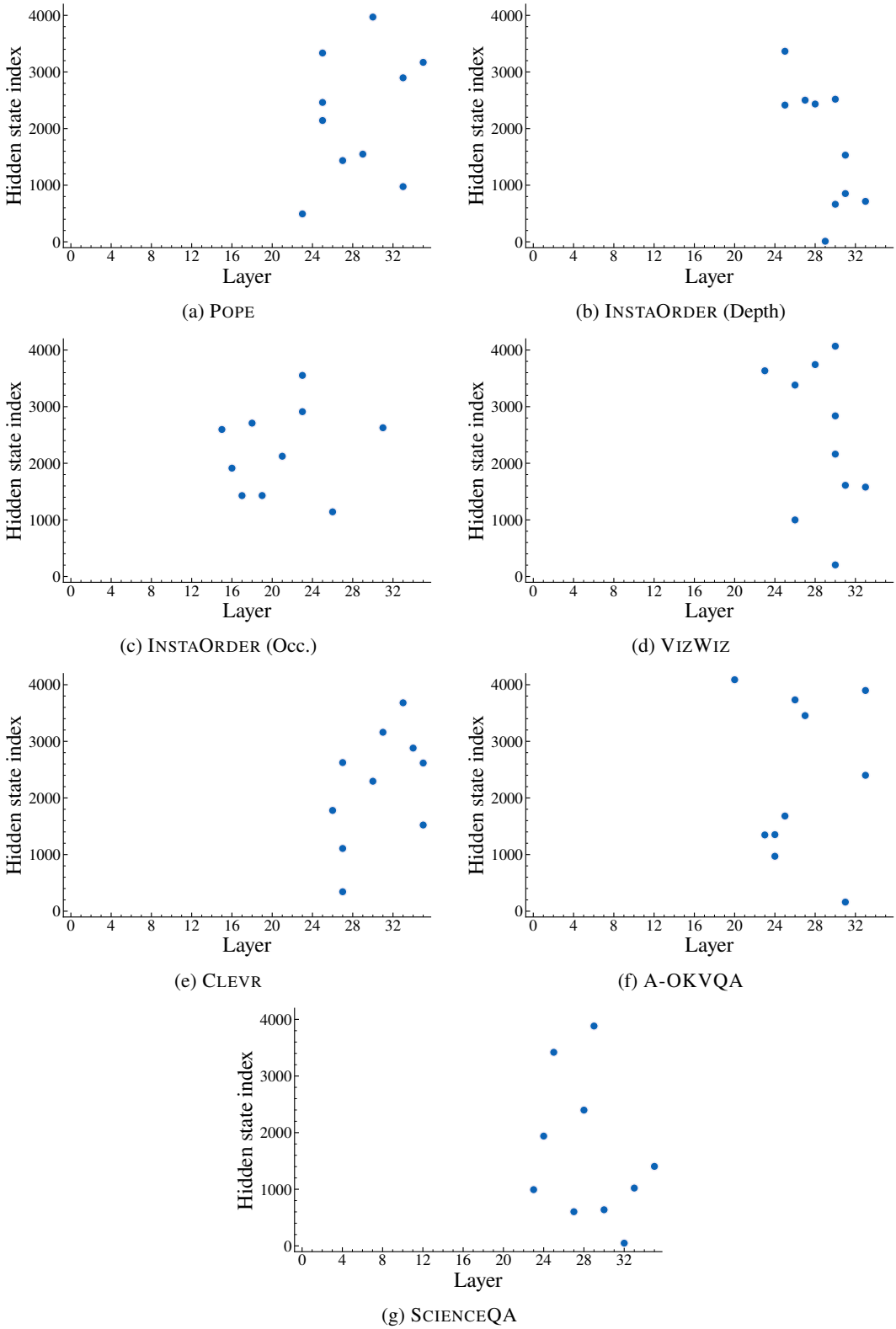

(a) POPE (b) INSTAORDER (Depth)

(c) INSTAORDER (Occ.) (d) VIZWIZ

(e) CLEVR (f) A-OKVQA

(g) SCIENCEQA

Figure 13: Location of the best $k = 10$ *super neurons* on LLaVA-v1.5-7b across all datasets.




(a) POPE (b) INSTAORDER (Depth)

(c) INSTAORDER (Occ.) (d) VIZWIZ

(e) CLEVR (f) A-OKVQA

(g) SCIENCEQA

Figure 14: Location of the best $k = 10$ *super neurons* on Qwen3-VL-8b-Instruct across all datasets.