Concept Learning for Interpreting Large Vision–Language Models

Md Abdul Kadir[1,2] **Omair Shahzad Bhatti**[1] **Daniel Sonntag**[1,2]

[1]German Research Center for Artificial Intelligence (DFKI), Saarbrücken, Germany
[2]University of Oldenburg, Oldenburg, Germany

{abdul.kadir, omair_shahzad.bhatti, daniel.sonntag}@dfki.de

Abstract

Large vision–language models (LVLMs) are inherently opaque, making it difficult to determine whether their outputs are grounded in visual evidence or driven by language-model priors. Existing concept-based interpretability methods are limited to classification settings, rely on proxy models, or require predefined tokens, and thus do not extend to LVLMs. We propose Text-Guided Concept Learning (TGCL), a weakly supervised framework for extracting multimodal concept vectors in LVLMs without token supervision. TGCL builds concept-to-image mappings from data and extracts patch-level activations via concept-guided probing. It then formulates concept learning as a contrastive disentanglement problem, isolating concept-specific patches from background patches to produce sparse, stable, and semantically aligned concept vectors. We conduct experiments on four datasets—ImageNet, MSCOCO, CIFAR100, and DTD—with three recent LVLMs. Across ImageNet and MSCOCO, TGCL consistently outperforms state-of-the-art baselines, improving concept sparsity, stability, overlap, attribution alignment, and faithfulness by up to 20, 8, 30, 20, and 4 percentage points, respectively. Code is available at TGCL-LVLM.

1 Introduction

Large vision–language models (LVLMs), such as LLaVA [1], Qwen [2], and Gemma [3], have demonstrated impressive capabilities in multimodal understanding and generation by combining visual encoders with large language models (LLMs). However, despite their strong performance, these models remain fundamentally opaque: it is often unclear whether their outputs are grounded in visual evidence or driven by language-model priors, leading to hallucinations and unreliable reasoning [4].

While concept-based methods have proven effective for image classifiers [5, 6, 7], extending them to LVLMs is non-trivial due to tokenization and autoregressiveness. We identify a key mismatch: *existing concept-based approaches assume a single, well-defined activation per input, whereas LVLMs produce autoregressive, multi-token activations with entangled visual and linguistic signals*. Consequently, concepts in LVLMs are rarely confined to a single activation; rather, they are distributed across multiple token activations and entangled with background visual cues and linguistic priors.

This mismatch introduces two fundamental challenges. First, it is unclear *which activations* should be used for concept extraction, as different token activations capture different aspects of the input. Second, LVLM activations exhibit *superposition* [8], where multiple concepts co-occur within the same image and are entangled in the activation space. Prior methods rely on predefined tokens and operate on full-image activations [9], limiting their ability to capture localized or co-occurring concepts.

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

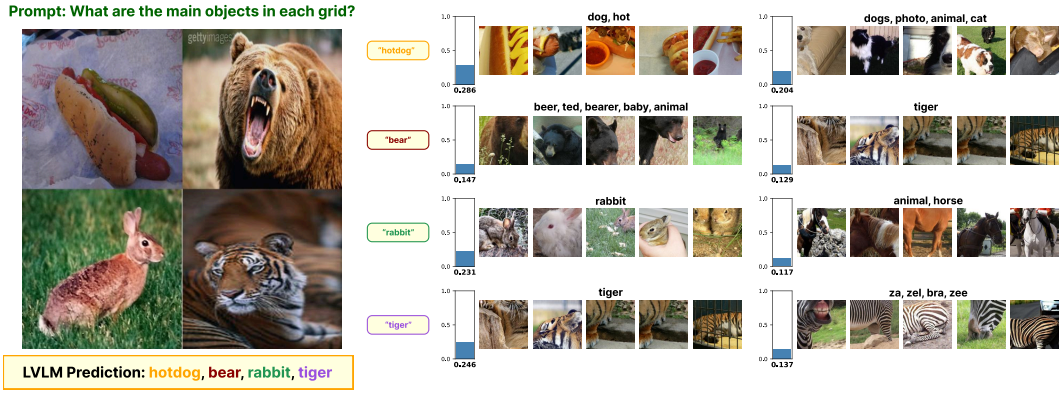


Figure 1: Example of TGCL attribution. Given a grid-structured image (left), Gemma-3n-E4B predicts the main objects in the grid, and TGCL selects the top two attributed concepts (right) from its learned concept dictionary. The corresponding image patches and decoded tokens show that predictions are grounded in semantically aligned visual and textual evidence. Additional examples are provided in Appendix F.3.

To address these limitations, we propose TGCL (Text-Guided Concept Learning), a framework for *token-free concept discovery and grounding in LVLMs*. The key idea is to shift concept extraction from token-level activations to *patch-level activations guided by textual concepts*, enabling direct alignment between visual regions and textual semantic descriptors. TGCL constructs a text-to-image mapping by prompting the LVLM to identify candidate textual concepts from an image collection. It then performs text-guided probing to extract patch-level activations from both concept-relevant and background regions via image cropping.

Crucially, we formulate concept learning as a *linear contrastive disentanglement problem*, in which concept-relevant signals are isolated from background noise using positive and negative patch samples. This yields sparse, stable, and semantically aligned concept vectors without requiring predefined tokens, proxy models, or architectural modifications. We keep the decomposition linear so that the learned concept vectors remain in the LVLM's activation space, where the output head can decode them (Sec. 3.3) and activations can be projected onto them linearly (Sec. 3.4). A nonlinear encoder would introduce additional parameters that are themselves unaudited, and its concept vectors would no longer lie in the model's native activation space. Furthermore, by operating at the patch level, TGCL naturally reduces inter-concept interference and mitigates the effects of autoregressive token entanglement [5].

Our main contributions are as follows:

- We introduce *TGCL*, a weakly supervised framework for scalable concept discovery in autoregressive LVLMs, overcoming the single-object limitation of prior methods.
- We propose a contrastive activation extraction scheme that encourages monosemanticity by separating concept-relevant vs. concept-irrelevant activations.
- We conduct experiments on four datasets—*ImageNet* [10], *MSCOCO* [11], *CIFAR100* [12], and *DTD* [13]—with three recent LVLMs. TGCL outperforms SOTA interpretability methods by up to 20 percentage points in sparsity, 8 in stability, 30 in concept overlap, 20 in attribution alignment, and 4.2 in insertion–deletion faithfulness.

2 Related Work

One of the main goals of machine learning interpretability is to uncover how internal representations encode semantically meaningful concepts in models. Early work introduced Concept Activation Vectors (CAVs) [5], which quantify model sensitivity to human-defined concepts but require curated examples, limiting scalability. Follow-up work sought to automate discovery [14], yet dependence on segmentation quality hindered robustness.

Some methods reduce supervision by identifying interpretable directions in image-patch activations. RONMF [15], CRAFT [6], MCD [16], UUCV [17], and SAE [18] have shown promise in vision-only settings within an unified framework [19]. CLIP-based concept attribution [7, 20] has further advanced automated concept discovery in CLIP's vision encoder, but these methods remain limited in their ability to interpret autoregressive LVLMs. Orthogonal approaches use saliency maps [21, 22, 23] or training-based grounding [24, 25, 26] to locate *where* evidence lies. While effective for localizing evidence, these methods do not clarify *what* abstract concepts a model internally represents, providing an orthogonal perspective that complements concept-based explanations.

Moreover, concept bottleneck models (CBMs) explain vision classifiers or image encoders by using an LVLM to extract textual concepts from images and then training a surrogate classification head on these concepts to predict a class [27]. In contrast, we use an LVLM to identify dataset-relevant concepts and leverage them to extract concept vectors from the same model's hidden states, enabling post-hoc explanation of its own predictions without training a surrogate classifier or modifying its prediction pathway.

Feature finding [18, 28] for understanding a model's knowledge and for model steering using sparse autoencoders is another promising direction for concept-level explanation. However, these methods introduce additional SAE-based representations rather than directly analyzing the model's original hidden-state dimensions, potentially altering its inference behavior and complicating strictly post hoc explanations. While concept extraction overlaps with mechanistic interpretability [29, 18, 8], our focus is on *single-layer monosemantic concept discovery* in autoregressive LVLMs rather than neuron- or circuit-level analysis. Alternative approaches, including attention maps [30], causal tracing [31], and linear probes [32], offer complementary perspectives on model behavior by examining information flow, causal dependencies, or representations in hidden states. However, they do not directly identify and characterize the dataset-relevant concepts captured by an LVLM for explaining its own predictions.

Autoregressive LVLMs present new challenges: activations evolve across time steps, residual streams encode multiple dependencies, and concepts rarely align with a single hidden state [29]. Parekh et al. [9] adapted Semi-NMF (SNMF) [33] to LVLM activations but struggles with (i) reliance on tokenized object names, (ii) limited support for multi-token concepts, (iii) background noise from full-image-based activation extraction, and (iv) persistent polysemanticity in residual streams. While methods such as TCAV, CRAFT, and CLIP-Dissect pioneered concept-based interpretability, they were designed for static encoders and are not directly architecturally suited to LVLMs. CoX-LMM [9] extends these ideas into a dictionary learning framework for autoregressive models and thus serves as the most representative baseline.

Unlike existing concept extraction methods that rely on labeled tokens, yield polysemantic vectors, and assume single-object dataset settings, TGCL introduces contrastive residual extraction with spatially localized image crops and candidate textual concepts. This encourages a clean separation between *concept* and background *noise*, producing faithful, monosemantic concept vectors. To the best of our knowledge, TGCL is the first framework to scale weakly supervised concept discovery from single-object to multi-object dataset settings.

3 Method

TGCL is model-agnostic and can be applied to any frozen LVLM f with accessible residual activations. Given such a model, TGCL works in four stages: constructing Positive and Negative Concept Bags (Section 3.1), applying Text-Guided Dictionary Learning for each concept (Section 3.2), identifying Positive Concept Vectors (Section 3.3), and finally using these vectors for Concept Attribution and Multi-modal Alignment (Section 3.4).

3.1 Positive and Negative Concept Bags

Positive and negative concept bags consist of image patches representing the presence and absence of a given concept. To build them, we follow two steps. First, we use the model f for automatic concept discovery (Figure 2 step 1) from an image collection using prompting. Although a secondary model could be used for this task, we use the same model to explain the concepts it understands. Second, foreground and background crops serve as positive and negative samples for each concept (Figure 2,

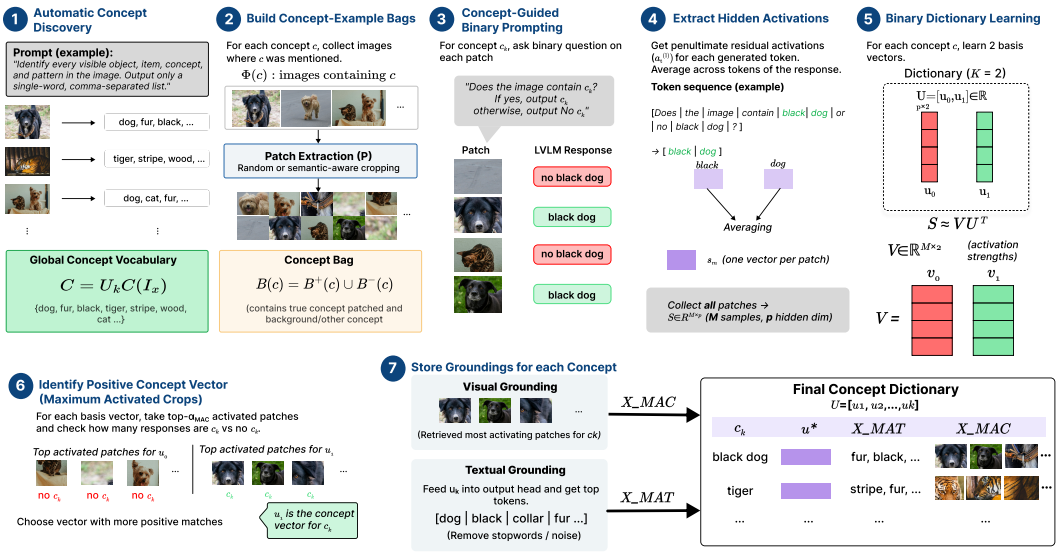


Figure 2: Overview of TGCL. TGCL first discovers a concept vocabulary and builds concept-example bags from unlabeled images. It then probes each crop with a binary concept prompt, extracts residual activations from the shared concept-token span, and learns a two-basis dictionary separating concept-present from concept-absent directions. The positive concept vector is chosen by selecting the direction whose maximally activating crops cause the model to generate the corresponding concept. This vector is then saved with its visual and textual groundings for post-hoc attribution.

step 2). Given unlabeled images $\mathcal{I} = \{I_k\}_{k=1}^N$, TGCL first prompts the LVLM to list visible objects, parts, attributes, textures, and patterns. This yields image-level candidate concepts.

$$C(I_k) = f(I_k, \textit{prompt}), \qquad \mathcal{C} = \bigcup_{k=1}^{N} C(I_k).$$

For each concept $c \in \mathcal{C}$, we collect supporting images.

$$\Phi(c) = \{I_k \mid c \in C(I_k)\}.$$

We then extract candidate crops from each supporting image using random crops or SAM proposals [34], denoted by $\mathcal{P}(I_k, c)$ (Figure 2 step 2). The resulting concept bag:

$$\mathcal{B}(c) = \{\, \mathcal{P}(I_k, c) : I_k \in \Phi(c) \,\} = \mathcal{B}^+(c) \cup \mathcal{B}^-(c),$$

where $\mathcal{B}^+(c)$ and $\mathcal{B}^-(c)$ are latent concept-present and concept-absent crops. TGCL does not assume these subsets are known; it separates them using concept-guided prompting in Section 3.2. This weakly supervised approach scales effectively to multi-object datasets and broader concepts—such as "stripes"—which appear across diverse classes like zebras, tigers, or cats without requiring manual annotations.

3.2 Text-Guided Dictionary Learning

Concept quality depends on which activations are decomposed, so we first extract concept-focused residuals with a contrastive prompt and then factorize them per concept. Steps 3-5 of Figure 2 describe the core TGCL learning procedure. Most concept-expansion methods rely on dictionary learning, but concept quality strongly depends on the activations passed to the decomposition [19, 16, 35].

Prior LVLM concept methods often rely on open-ended generation and single-token analysis, where residuals are influenced by the full sequence and may mix objects, attributes, and language priors. TGCL instead uses constrained textual concept generation to reduce noise and entanglement. This leads to overlapping concepts [29] and prevents dictionary elements from representing clean, disentangled semantics. Multi-object images further exacerbate this effect, as activations mix features from different objects.

We address these issues with *Concept-Guidance*, a contrastive residual extraction scheme. The model is prompted to output either the target concept c_k or *No* c_k, encouraging activations to align with a single concept and yielding cleaner, concept-focused residuals:

> $\text{prompt}_{\text{cg}}$ = "*Does the image contain* c_k*? If yes, output* c_k*; otherwise, output No* c_k."

For each concept bag $\mathcal{B}(c_k)$, corresponding to the k-th textual concept in $\mathcal{C}$, we query the model with the above prompt and collect residual activations. Following Koh et al. [36], Alam et al. [37], we decompose the LVLM into an embedding function g and an output function h. Based on Fel et al. [19], Parekh et al. [9], we analyze the penultimate residual layer. For an image crop $x^{c_k} \in \mathcal{B}(c_k)$ and cached prefix $\hat{y}_{<t}$, the embedding function produces

$$a_t^{(l)} = g^{(l)}(x^{c_k}, \textit{prompt}_{cg}, \hat{y}_{<t}) \in \mathbb{R}^p, \tag{1}$$

where p is the residual dimension [38]. The output function predicts

$$\hat{y}_t = h^{(l)}(a_t^{(l)}). \tag{2}$$

A key detail is that TGCL averages residuals only over the shared concept-token span. For a positive response such as *hot dog*, we pool over the tokens corresponding to *hot dog*. For a negative response such as *No hot dog*, we discard the negation token *No* and again pool only over the shared span *hot dog*. Thus, positive and negative samples use the same textual span, and the contrast comes from image-conditioned residual differences rather than token identity.

Let $\mathcal{T}_{m,c_k}$ denote the token positions corresponding to the concept span c_k in sample m. If there are multiple tokens, we average their activations, following standard practice for combining multiple representations [39, 38, 40].

$$s_m^{c_k} = \frac{1}{|\mathcal{T}_{m,c_k}|} \sum_{t \in \mathcal{T}_{m,c_k}} a_t^{(l)} \in \mathbb{R}^p. \tag{3}$$

Collecting M_k such samples yields

$$S_{c_k} = [s_1^{c_k}, \ldots, s_{M_k}^{c_k}]^\top \in \mathbb{R}^{M_k \times p}, \tag{4}$$

which serves as input for concept extraction. Unlike approaches that rely solely on single-token embeddings, this formulation captures information from multi-token concepts, e.g., *hot dog*, rather than splitting them into isolated tokens.

Dictionary learning then decomposes S_{c_k} into concept and negation bases:

$$S_{c_k} \approx V_{c_k} U_{c_k}^\top, \quad V_{c_k} \in \mathbb{R}^{M_k \times 2}, \quad U_{c_k} \in \mathbb{R}^{p \times 2}.$$

U_{c_k} contains two basis vectors for c_k and *No* c_k, while V_{c_k} capturing sample activations. When restricting each bag to two bases, concept-present vs. concept-absent, the per-bag decomposition cost remains small, and the method avoids interference among many dictionary atoms. Over K concepts, TGCL learns K independent two-basis dictionaries, which scales naturally to large concept sets and remains consistent with the per-iteration complexity of Semi-NMF [41, 42]. This contrasts with large joint dictionaries, where many atoms can interfere or collapse into overlapping concepts, and with SAE, where training requires backpropagation through additional parameters. This formulation contrasts concept-present and concept-absent residuals within each concept bag, yielding concept-specific directions without requiring oversize dictionaries. Unlike the prior method [9], which extracts entangled activation from token sequences, text-guidance produces more controlled activation suitable for LVLM interpretability.

3.3 Positive Concept Vector Identification

The two learned bases are unlabeled, so we must decide which one encodes the concept and attach visual and textual evidence to it. We identify the positive concept direction using maximum-activated patches (MAC) (Figure 2 step 6). For each basis $u_j \in U_{c_k}$, we retrieve the top-α_{MAC} crops according to the activation coefficients $(V_{c_k})_{:,j}$, denoted $X_{c_k,j,\text{MAC}}$. It is selected automatically based on the

number of images per concept c, where each image contributes at least one concept instance, yielding up to c positive samples in total. Since each concept bag contains both c_k and *No* c_k samples, we select the basis whose maximum-activated crops are most often predicted by the LVLM as containing c_k:

$$j^* = \arg\max_{j \in \{1,2\}} \left| \{ i \in X_{c_k,j,\mathrm{MAC}} : \hat{y}_i = c_k \} \right|.$$

We define the positive concept vector as $u_{c_k} = U_{c_k}[:, j^*]$ and its visual grounding as $X_{c_k,\mathrm{MAC}} = X_{c_k,j^*,\mathrm{MAC}}$. Repeating this for all concepts and symmetrically orthogonalizing them yields the TGCL dictionary $\widehat{U} = [u_{c_1}, \ldots, u_{c_K}] \in \mathbb{R}^{p \times K}$ with visual groundings $X_{\mathrm{MAC}} = \{X_{c_1,\mathrm{MAC}}, \ldots, X_{c_K,\mathrm{MAC}}\}$.

For textual grounding, we apply the LVLM output head to each concept vector, $q_{c_k} = h^{(l)}(u_{c_k}) \in \mathbb{R}^{|\mathcal{V}_{\mathrm{tok}}|}$, select the top-$\tau$ tokens, and remove stopwords and noisy subword fragments. The resulting token set $X_{c_k,\mathrm{MAT}}$ provides a textual summary of the concept direction; collecting these sets gives $X_{\mathrm{MAT}} = \{X_{c_1,\mathrm{MAT}}, \ldots, X_{c_K,\mathrm{MAT}}\}$.

3.4 Concept Attribution and Multi-modal Alignment

With the dictionary fixed, we explain an LVLM output by measuring how strongly its activations align with each concept vector. Given the TGCL dictionary $\widehat{U} = [u_{c_1}, \ldots, u_{c_K}]$, we attribute LVLM predictions by projecting residual activations at a layer l onto the learned concept subspace [5]. At the token level, the attribution score of a concept c_j for the generated token t is given by $\alpha_j^{(t)} = \cos_sim(u_{c_j}, a_t^{(l)})$. For phrase-level attribution, let $\mathcal{T}_m$ denote the set of token positions of the generated phrase or response span in sample m, analogous to $\mathcal{T}_{m,c_k}$ in Eq. (3). We use the mean-pooled activation $s_m = \frac{1}{|\mathcal{T}_m|} \sum_{t \in \mathcal{T}_m} a_t^{(l)}$ and compute $\alpha_{m,j} = \cos_\mathrm{sim}(u_{c_j}, s_m)$.

4 Experiments

We evaluate our concept extraction method on three recent LVLMs: *Gemma-3n-E4B* [3], *Qwen2-VL-7B* [2], and *Qwen2.5-VL-7B* [43], comparing it against three SOTA concept extraction methods. Detailed descriptions of the models are provided in Appendix D. Further, TGCL is evaluated on four diverse datasets: ImageNet [10], MSCOCO [11], DTD [13], and CIFAR-100 [12]. These datasets are chosen to assess concept extraction under different visual settings: single-object images (ImageNet), multi-object scenes (MSCOCO), texture and pattern-based data (DTD), and low-resolution images (CIFAR-100). All evaluations are conducted on a separate validation set, distinct from the datasets used for concept extraction by the baseline methods. Further details of the datasets are provided in Appendix E.

4.1 Experiments and Metrics

TGCL extracts concept vectors and performs attribution using concept examples. Accordingly, we divide our experiments into three parts: *concept vector quality analysis*, *attribution quality evaluation*, and *concept faithfulness*.

Concept Vector Quality Analysis. To assess the quality of the discovered concepts, we use three standard metrics, as defined in recent work [19, 9]: **Sparsity** (↑), the fraction of near-zero entries in the extracted concept vectors $\widehat{U}$, which indicates how few neuron activations represent each concept vector; **Instability** (↓), the misalignment between concept vectors learned across different runs (in the experiment we used 5 runs), computed via Hungarian matching based on cosine similarity between concept vectors; and **Overlap** (↓), the average pairwise cosine similarity between distinct concept vectors, capturing redundancy in the learned concept basis. We compare our results with recent concept extraction methods, including CRAFT [6], MCD [16], and CoX_LMM [9]. For concept decomposition, we evaluate three decomposition approaches: SAE [19], SNMF [9], and SIMPLE [9]. SIMPLE decomposes features using k-means clustering and defines activation projections based on the distance between features and cluster centers.

Attribution Quality Evaluation. For attribution quality analysis, we pass a 2×2 grid of images (Figure 1) to the LVLM and ask the model to predict the objects present in each grid cell. We then compute the attribution of each prediction to the corresponding concepts. Instead of using a single image, we use a grid of unrelated images to introduce greater difficulty in concept attribution and compare attribution quality under this setting. Attribution quality is evaluated using **CLIPScore** [44] and **BERTScore** [45], which are standard metrics for text-image and text-text alignment, respectively. CLIPScore measures cross-modal contrastive alignment between model predictions and attributed visual concepts, while BERTScore captures semantic similarity between model predictions and attributed textual concepts.

Concept Faithfulness. Concept faithfulness evaluates whether the importance ranking of extracted concepts aligns with their actual influence on the model prediction. In particular, concepts that are assigned higher attribution scores should correspond to larger effects on the predicted token distribution when perturbed. For example, removing highly influential concept information from the internal representation should lead to a larger decrease in the predicted token probability compared to less relevant concepts. Faithfulness is evaluated using concept insertion and deletion curves [19, 46], which measure changes in model output as influential concepts are progressively removed or added. We report results for the top-1 to top-3 concepts ranked by their impact on the model prediction. In C-Deletion, we progressively remove the most influential concept directions (ranked by gradient magnitude with respect to the highest-probability token) and measure the drop in output probability. In C-Insertion, we start from a zero vector and incrementally add the same concept directions, tracking the corresponding probability increase. Furthermore, we present qualitative examples of the extracted concepts and concept attribution outputs for all three LVLMs on the ImageNet and DTD datasets and compare our results with SOTA methods.

4.2 Results

The *concept vector quality analysis* for Gemma-3n-E4B on ImageNet and MSCOCO is shown in Table 1. We observe that CoX-LMM and TGCL perform better than the other two concept extraction methods on both datasets. We also notice that TGCL with SNMF consistently achieves the best performance, with the highest sparsity, lowest instability, and lowest overlap. In the next experiments, we therefore compare TGCL with the SOTA using SNMF decomposition.

Table 1: Concept vector evaluation on ImageNet and MSCOCO using Gemma-3n-E4B LVLM with state-of-the-art concept extraction methods, compared with our approach across different concept decomposition methods (SNMF, SAE, and SIMPLE). Higher sparsity (↑) and lower instability and overlap (↓) indicate better performance.

Method	**Decomp.**	**ImageNet**			**MSCOCO**		
		Sparsity↑	Instability↓	Overlap↓	Sparsity↑	Instability↓	Overlap↓
CRAFT	SNMF	0.42	0.02	0.69	0.42	**0.00**	0.81
	SAE	0.39	0.55	0.02	0.37	0.81	0.04
	SIMPLE	0.42	**0.00**	0.52	0.42	0.33	0.59
MCD	SNMF	0.42	0.02	0.70	0.42	**0.00**	0.78
	SAE	0.39	0.67	0.02	0.37	0.78	0.05
	SIMPLE	0.42	**0.00**	0.67	0.43	0.23	0.45
CoX-LMM	SNMF	0.43	0.06	0.27	0.43	**0.00**	0.32
	SAE	0.42	0.80	0.02	0.38	0.86	0.03
	SIMPLE	0.44	**0.00**	0.34	0.43	0.09	0.36
TGCL (ours)	SNMF	**0.61**	**0.00**	**0.01**	**0.63**	0.01	**0.02**
	SAE	0.36	0.96	0.78	0.42	1.00	0.04
	SIMPLE	0.45	**0.00**	0.53	0.44	0.01	0.68

We perform attribution quality evaluation on Gemma-3n-E4B, Qwen2-VL-7B, and Qwen2.5-VL-7B across ImageNet, MSCOCO, DTD, and CIFAR-100. Table 2 reports attribution results for Gemma-3n-E4B on ImageNet and MSCOCO. Results for DTD and CIFAR-100, as well as for Qwen2-VL-7B and Qwen2.5-VL-7B, are provided in Appendix F. We report results under three settings: **Text-only**, where LVLM activations are probed using class names and the most strongly attributed concept is

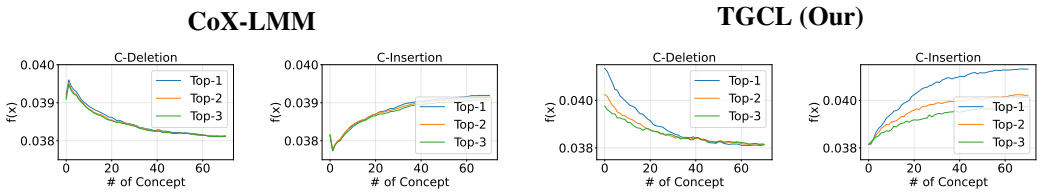


Figure 3: Faithfulness comparison of the SOTA and TGCL using concept deletion and insertion. CoX-LMM yields relatively flatter curves with weak separation across ranks, whereas TGCL preserves a clear order (Top-1 >Top-2 >Top-3), with sharper degradation under deletion and stronger recovery under insertion. This shows that TGCL produces more faithful and discriminative concept rankings.

selected; **Image-only**, where the LVLM first predicts visual descriptions and concept attribution is performed based on these predictions; and **Combined**, where the LVLM predicts all objects in a 2×2 image grid, and attribution is computed based on these predictions of each grid item. Across both datasets and all evaluation metrics, TGCL consistently outperforms prior methods, with gains of approximately 5% in the text-only setting, 6% in the image-only setting, and up to 20% in the combined setting over the strongest baseline. Random baselines—constructed by assigning concepts randomly rather than using ranked attribution—yield near-constant scores and consistently achieve the lowest CLIPScore and BERTScore, confirming that meaningful concept alignment is necessary for strong performance.

Table 2: Comparison of TGCL and CoX-LMM on CLIPScore and BERTScore across datasets in unimodal and multimodal settings of LVLMs. Higher metric values indicate better attribution quality, while **Random** represents the baseline score for both metrics. The table suggests that TGCL performs better in the image-only and combined settings on both CLIPScore and BERTScore across datasets, while CoX-LMM performs relatively low in comparison.

Method	Dataset	Concept	Metric	Random	Text-only	Image-only	Combined
TGCL (Our)	ImageNet	1,000	CLIPScore	0.52 ± 0.04	–	**0.62 ± 0.08**	**0.67 ± 0.09**
			BERTScore	0.71 ± 0.06	0.78 ± 0.07	0.84 ± 0.08	0.86 ± 0.10
	MSCOCO	10	CLIPScore	0.48 ± 0.04	–	0.60 ± 0.06	0.64 ± 0.08
			BERTScore	0.71 ± 0.01	**0.89 ± 0.06**	**0.91 ± 0.05**	**0.93 ± 0.07**
CoX-LMM	ImageNet	1,000	CLIPScore	0.49 ± 0.04	–	0.57 ± 0.03	0.58 ± 0.05
			BERTScore	0.71 ± 0.00	0.74 ± 0.08	0.75 ± 0.06	0.82 ± 0.09
	MSCOCO	10	CLIPScore	0.51 ± 0.04	–	0.57 ± 0.10	0.55 ± 0.05
			BERTScore	0.71 ± 0.01	0.83 ± 0.01	0.79 ± 0.09	0.73 ± 0.11

Concept faithfulness comparison for CoX-LMM and TGCL is evaluated on MSCOCO using standard faithfulness metrics, namely concept deletion (C-Deletion) and insertion (C-Insertion), with Gemma-3n-E4B. For each image, the model is prompted for main object classification, and the top-3 concepts are attributed using residual activations (Sec. 3.4). Scores are averaged over tokens and aggregated over the validation set. As shown in Figure 3, TGCL exhibits a clear ranking, where higher-ranked concepts induce larger changes in both insertion and deletion, indicating faithful attribution. In contrast, CoX-LMM shows weak separation across ranks, with nearly uniform influence of top concepts. Overall, TGCL produces more faithful and discriminative concept rankings. We also present the numerical evaluation of concept faithfulness in Table 3, demonstrating higher insertion AUC and lower deletion AUC compared to SOTA methods.

We visualize the concept examples extracted from Gemma-3n-E4B in Figure 4, which compares ImageNet concepts extracted by CoX-LMM (left) and TGCL (right). TGCL yields fine-grained, monosemantic representations (e.g., fur, stripes), whereas CoX-LMM produces entangled groundings that mix semantics (e.g., tiger conflated with lion or multiple animals). Furthermore, Figure 1 presents concept attribution of Gemma-3n-E4B using TGCL, where the left panel shows the input image, the middle panel shows the model output, and the right panel shows the most attributed results using the top-2 concepts. We provide additional examples for Qwen2-VL-7B and Qwen2.5-VL-7B on

Table 3: Concept faithfulness on MSCOCO-10 (Gemma-3n-E4B). Mean ± std over 500 images. Gap = Ins − Del. *Random order*: the insertion and deletion order is randomized. Δ is TGCL (SAM) minus the row (Welch's t-test, n=500 per group, Holm-corrected within each family, insertion and deletion tested separately; all $p < 0.001$; Cohen's d in brackets). Rank spread is the difference between the largest and smallest gap across concept ranks; a small spread means the ranking carries no ordinal information. TGCL gap by rank: $0.650/0.554/0.542$ (SAM), $0.628/0.552/0.542$ (random crops).

	Ranked order			Random order			TGCL (SAM) advantage		
Method	Ins ↑	Del ↓	Gap ↑	Ins	Del	Gap	ΔIns (d)	ΔDel (d)	Rank spread ↑
CRAFT	0.781±0.049	0.217±0.053	0.564	0.510±0.005	0.491±0.004	0.019	+0.045 (1.28)	−0.041 (−1.08)	0.016
MCD	0.793±0.046	0.185±0.033	0.608	0.512±0.004	0.489±0.004	0.023	+0.033 (1.00)	−0.009 (−0.37)	0.016
CoX-LMM	0.771±0.018	0.194±0.015	0.577	0.512±0.005	0.488±0.005	0.024	+0.055 (3.87)	−0.018 (−1.50)	0.016
TGCL (random crops)	0.815±0.011	0.187±0.009	0.628	0.517±0.010	0.489±0.008	0.028	+0.011 (1.09)	−0.011 (−1.29)	0.086
TGCL (SAM)	**0.826**±0.009	**0.176**±0.008	**0.650**	0.516±0.011	0.479±0.009	0.037	ref.	ref.	**0.108**

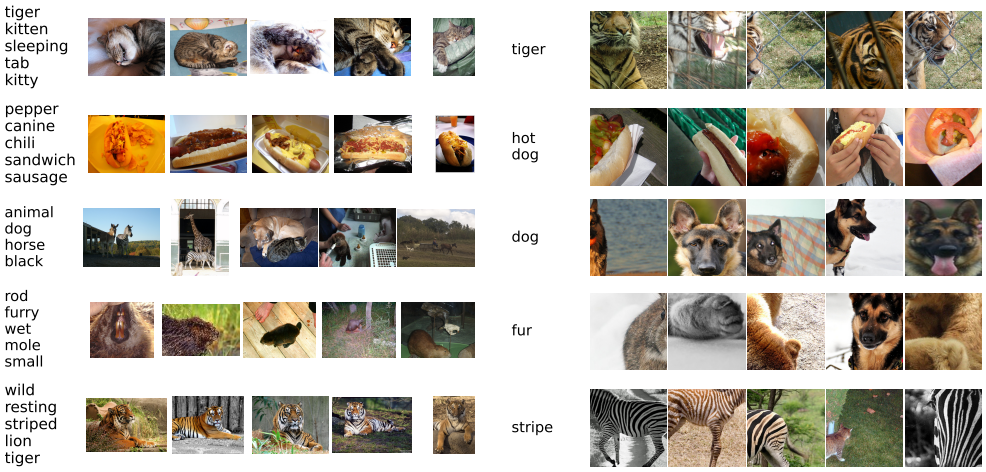


Figure 4: Qualitative examples of concept representations extracted from ImageNet. CoX-LMM (left) concepts are often grounded in multiple unrelated or overlapping tokens (e.g., *"canine"* and *"hot dog"* mapped to the same concept), reflecting polysemantic vectors. In several cases, concepts mix distinct animals: for example, *"tiger"* grounds across multiple concepts, while *"lion"* is misrepresented as *"tiger"*. Such cases illustrate non-monosemantic behavior, where concept vectors capture blended rather than distinct semantics. In contrast, TGCL (right) discovers fine-grained, monosemantic concepts (e.g., *fur*, *dog*, *stripes*).

ImageNet and DTD in Appendix F.3–F.5. Appendix F.4 (Figure 8) shows that when an object's corresponding concept is absent from the dictionary, TGCL attributes the prediction to semantically related concepts, such as carriage → horse. Appendix F.5 (Figure 11) demonstrates that concepts learned from DTD textures can also explain ImageNet predictions through abstract visual concepts, such as macaw → colorful and ladybug → polka-dot.

5 Ablation Study

We test multiple variants of the contrastive prompt template (see Appendix C for details). Table 5 shows only minor fluctuations in CLIPScore, indicating robustness to phrasing. Whereas Qwen2-VL-7B exhibits slightly higher variance than Qwen2.5-VL-7B and Gemma-3n-E4B, which could be due to its limited performance. Furthermore, we ablate the SNMF sparsity weight α (Table 10) and decomposition components $K > 2$ (Table 9) in Appendix G, and find that attribution results remain stable, while CLIP scores decrease for large K. As the number of dictionary decompositions increases, they tend to capture more abstract representations, and the CLIP score can decrease because the ground-truth text does not adequately describe these abstract and fine-grained concepts. Hence, we select $\alpha = 20$ and $K = 2$, which we find sufficient for attribution.

We also compare semantic-aware cropping using SAM with a simpler random-cropping baseline for visual concept extraction (Table 11 in Appendix G). SAM yields slightly higher BERTScore and CLIPScore, as its segmentation gives better semantic representation than random crops. Detailed numerical results are reported in Appendix G, Table 11, and qualitative examples are shown in Figures 8 and 6. Further, in Appendix G we investigated which layer yields better attribution quality in terms of BERTScore and CLIPScore. We observe that CLIPScore generally increases in deeper layers, suggesting better human-understandable concepts and alignment with the corresponding images. We provide an additional ablation in Appendix G showing that, even with randomly sampled crops, performance improves consistently with bag size and saturates near 900 per concept. This indicates that explicit control of the positive-to-negative crop ratio is unnecessary when the bag size is sufficiently large, as the data already provides an implicit balance through dataset-specific structure. In particular, each image contains at least one positive concept instance, which the LVLM identifies during automatic concept discovery.

Table 4: Evaluation of concept validity on a fixed 10-category COCO vocabulary. Contrastive prompting and concept-span pooling are evaluated in a 2×2 ablation design, where the shuffled-concept row acts as a random control. Best values are shown in bold.

Prompt	Pooling	Overlap ↓	Instab. ↓	BERT ↑	CLIP ↑	F1 ↑
Contrastive	concept span	**0.06**	0.00102	**0.936**	**0.618**	0.857
Contrastive	full output	0.09	**0.00095**	0.901	0.597	**0.858**
Non-contrastive	concept span	0.17	0.02118	0.729	0.556	0.351
Non-contrastive	full output	0.66	0.20256	0.714	0.523	0.314
Shuffled concept names	–	0.84	0.19260	0.702	0.518	0.012

Concept validity. By design, the LVLM proposes the concepts itself, so that TGCL can show what the model actually represents, including its biases. This raises a question: do the extracted concept vectors and their image evidence represent the textual concept, or a bias of the model that proposed it? To test this independently of the model, we fix the vocabulary to the 10 COCO categories and score concept vectors using the F1 score between the held-out ground truth and the rank-1 attributions. We also ablate the contrastive prompt and concept-span pooling in a 2×2 design and add a shuffled-concept control as a floor (Table 4). The contrastive prompt substantially improves F1 from 0.351 to 0.857 while reducing overlap and instability, whereas the shuffled control reduces F1 to 0.012, providing strong evidence that the learned vectors capture the intended concept rather than merely reflecting changes in the model's predictions. We also evaluated TGCL's computational performance (see Appendix H) and found that it is approximately $10\times$ faster than recent state-of-the-art methods for 10-concept extraction due to reduced FLOPs.

6 Conclusion

We introduce *Text-Guided Concept Learning* (TGCL), a weakly supervised framework that encourages monosemanticity and grounds concepts directly within LVLMs, yielding faithful multimodal concept attribution. Across three experimental settings—concept vector quality, concept attribution quality, and concept attribution faithfulness—TGCL consistently outperforms existing methods. We further conduct a comprehensive ablation study, ranging from prompt variations to the number of decomposition components in SNMF, and a computational cost comparison against SOTA methods. In future work, we aim to extend TGCL towards hierarchical concept learning.

Limitations. TGCL does not model hierarchical relations between concepts and requires the LVLM to follow basic language prompts, which holds for most modern LVLMs. Concepts are validated against independent references (CLIPScore, BERTScore, and concept-based classification F1 against class names). In the open-vocabulary setting, however, the LVLM proposes the concepts, judges their presence, and is the model being explained, so no independent reference exists, and its errors, biases, or hallucinations can propagate into the dictionary. Faithfulness is assessed with insertion/deletion against a random-order control; since zeroing concept directions may move activations off the data manifold, absolute probability changes may partly reflect distribution shift, but the gap relative to random order, which applies the same amount of zeroing, reflects the concept ranking baseline.

Acknowledgments and Disclosure of Funding

We thank our colleagues in Interactive Machine Learning (IML) at DFKI, Hasan Md Tusfiqur Alam, Abdulrahman Mohamed Selim, Tuan Tran, László Kopácsi, and Michael Barz, for their valuable feedback and insightful discussions.

This research was funded by the German Federal Ministry of Research, Technology and Space (BMFTR) under grant numbers 16IW23002 (No-IDLE), 16IW24006 (NoIDLEChatGPT), and 16IS23064 (IUML), as well as by the Lower Saxony Ministry of Science and Culture (MWK) through the zukunft.niedersachsen program and the Endowed Chair of AAI at the University of Oldenburg.

References

[1] Haotian Liu, Chunyuan Li, Qingyang Wu, et al. Visual Instruction Tuning. In *Advances in Neural Information Processing Systems*, volume 36, pages 34892–34916. Curran Associates, Inc., 2023. doi: 10.52202/075280-1516. URL https://dl.acm.org/doi/10.5555/3666122.3667638.

[2] Peng Wang, Shuai Bai, Sinan Tan, et al. Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution. arXiv preprint arXiv:2409.12191, October 2024. URL http://arxiv.org/abs/2409.12191.

[3] Gemma Team, Aishwarya Kamath, Johan Ferret, et al. Gemma 3 Technical Report. arXiv preprint arXiv:2503.19786, March 2025. URL http://arxiv.org/abs/2503.19786.

[4] Yifan Li, Yifan Du, Kun Zhou, et al. Evaluating Object Hallucination in Large Vision-Language Models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 292–305, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.20. URL https://aclanthology.org/2023.emnlp-main.20/.

[5] Been Kim, Martin Wattenberg, Justin Gilmer, et al. Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). In *Proceedings of the 35th International Conference on Machine Learning*, pages 2668–2677. PMLR, July 2018. URL https://proceedings.mlr.press/v80/kim18d.html.

[6] Thomas Fel, Agustin Picard, Louis Bethune, et al. CRAFT: Concept Recursive Activation FacTorization for Explainability. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2711–2721, June 2023. doi: 10.1109/CVPR52729.2023.00266. URL https://ieeexplore.ieee.org/document/10205223.

[7] Tuomas Oikarinen and Tsui-Wei Weng. CLIP-Dissect: Automatic Description of Neuron Representations in Deep Vision Networks. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=iPWiwWHc1V.

[8] Nelson Elhage, Tristan Hume, Catherine Olsson, et al. Toy Models of Superposition. arXiv preprint arXiv:2209.10652, September 2022. URL http://arxiv.org/abs/2209.10652.

[9] Jayneel Parekh, Pegah Khayatan, Mustafa Shukor, et al. A Concept-Based Explainability Framework for Large Multimodal Models. In *Advances in Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=MvjLRFntW6.

[10] Jia Deng, Wei Dong, Richard Socher, et al. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, June 2009. doi: 10.1109/CVPR.2009.5206848. URL https://ieeexplore.ieee.org/document/5206848.

[11] Tsung-Yi Lin, Michael Maire, Serge Belongie, et al. Microsoft COCO: Common Objects in Context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10602-1. doi: 10.1007/978-3-319-10602-1_48.

[12] Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Technical report, University of Toronto, 2009. URL https://www.semanticscholar.org/paper/Learning-Multiple-Layers-of-Features-from-Tiny-Krizhevsky/5d90f06bb70a0a3dced62413346235c02b1aa086.

[13] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, et al. Describing Textures in the Wild. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3613, June 2014. doi: 10.1109/CVPR.2014.461. URL https://ieeexplore.ieee.org/document/6909856.

[14] Amirata Ghorbani, James Wexler, James Y. Zou, et al. Towards automatic concept-based explanations. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 9277–9286. Curran Associates Inc., Red Hook, NY, USA, December 2019. URL https://dl.acm.org/doi/10.5555/3454287.3455119.

[15] Jingjing Liu, Nian Wu, Xianchao Xiu, et al. Robust orthogonal NMF with label propagation for image clustering. *Neural Networks*, 198:108613, June 2026. ISSN 0893-6080. doi: 10.1016/j.neunet.2026.108613. URL https://www.sciencedirect.com/science/article/pii/S0893608026000754.

[16] Arne Grobrügge, Niklas Kühl, Gerhard Satzger, et al. Towards Human-Understandable Multi-Dimensional Concept Discovery. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20018–20027, June 2025. doi: 10.1109/CVPR52734.2025.01864. URL https://ieeexplore.ieee.org/abstract/document/11093657.

[17] Mara Graziani, Laura O'Mahony, An-phi Nguyen, et al. Uncovering Unique Concept Vectors through Latent Space Decomposition. *Transactions on Machine Learning Research*, August 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=LT4DXqUJTD.

[18] Mateusz Pach, Shyamgopal Karthik, Quentin Bouniot, et al. Sparse Autoencoders Learn Monosemantic Features in Vision-Language Models. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 95706–95742. Curran Associates, Inc., 2025. doi: 10.52202/085713-3202. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/89e83382abeee53b932a6df62edbf9cc-Abstract-Conference.html.

[19] Thomas Fel, Victor Boutin, Louis Béthune, et al. A Holistic Approach to Unifying Automatic Concept Extraction and Concept Importance Estimation. In *Advances in Neural Information Processing Systems*, volume 36, pages 54805–54818. Curran Associates, Inc., 2023. doi: 10.52202/075280-2391. URL https://dl.acm.org/doi/10.5555/3666122.3668513.

[20] Maximilian Dreyer, Jim Berend, Tobias Labarta, et al. Mechanistic understanding and validation of large AI models with SemanticLens. *Nature Machine Intelligence*, 7(9):1572–1585, September 2025. ISSN 2522-5839. doi: 10.1038/s42256-025-01084-w. URL https://www.nature.com/articles/s42256-025-01084-w.

[21] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, et al. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, October 2017. doi: 10.1109/ICCV.2017.74. URL https://ieeexplore.ieee.org/document/8237336.

[22] Erik Štrumbelj and Igor Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems*, 41(3):647–665, December 2014. ISSN 0219-3116. doi: 10.1007/s10115-013-0679-x. URL https://doi.org/10.1007/s10115-013-0679-x.

[23] Fabrizio Nunnari, Md Abdul Kadir, and Daniel Sonntag. On the Overlap Between Grad-CAM Saliency Maps and Explainable Visual Features in Skin Cancer Images. In Andreas Holzinger, Peter Kieseberg, A. Min Tjoa, and Edgar Weippl, editors, *Machine Learning and Knowledge Extraction*, pages 241–253, Cham, 2021. Springer International Publishing. ISBN 978-3-030-84060-0. doi: 10.1007/978-3-030-84060-0_16. URL https://link.springer.com/chapter/10.1007/978-3-030-84060-0_16.

[24] Weitai Kang, Gaowen Liu, Mubarak Shah, et al. SegVG: Transferring Object Bounding Box to Segmentation for Visual Grounding. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision – ECCV 2024*, pages 57–75, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-72920-1. doi: 10.1007/978-3-031-72920-1_4.

[25] Hao Zhang, Hongyang Li, Feng Li, et al. LLaVA-Grounding: Grounded Visual Chat with Large Multimodal Models. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision – ECCV 2024*, pages 19–35, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-72775-7. doi: 10.1007/978-3-031-72775-7_2.

[26] Chuofan Ma, Yi Jiang, Jiannan Wu, et al. Groma: Localized Visual Tokenization for Grounding Multimodal Large Language Models. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision – ECCV 2024*, pages 417–435, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-72658-3. doi: 10.1007/978-3-031-72658-3_24.

[27] Tuomas Oikarinen, Subhro Das, Lam M. Nguyen, et al. Label-free Concept Bottleneck Models. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=FlCg47MNvBA.

[28] Kaichen Zhang, Yifei Shen, Bo Li, et al. Large Multi-modal Models Can Interpret Features in Large Multi-modal Models. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3650–3661, October 2025. doi: 10.1109/ICCV51701.2025.00348. URL https://ieeexplore.ieee.org/abstract/document/11444372.

[29] Adly Templeton, Tom Conerly, Jonathan Marcus, et al. Scaling Monosemanticity: Extracting Interpretable Features from Claude 3 Sonnet. arXiv preprint arXiv:2605.29358, May 2026. URL http://arxiv.org/abs/2605.29358.

[30] Sarthak Jain and Byron C. Wallace. Attention is not Explanation. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1357. URL https://aclanthology.org/N19-1357/.

[31] Kevin Meng, David Bau, Alex Andonian, et al. Locating and Editing Factual Associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35, pages 17359–17372. Curran Associates, Inc., 2022. doi: 10.52202/068431-1262. URL https://dl.acm.org/doi/10.5555/3600270.3601532.

[32] Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. In *International Conference on Learning Representations (ICLR)*, 2017. URL https://openreview.net/forum?id=HJ4-rAVtl.

[33] George Trigeorgis, Konstantinos Bousmalis, Stefanos Zafeiriou, et al. A Deep Semi-NMF Model for Learning Hidden Representations. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1692–1700. PMLR, June 2014. URL https://proceedings.mlr.press/v32/trigeorgis14.html.

[34] Alexander Kirillov, Eric Mintun, Nikhila Ravi, et al. Segment Anything. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3992–4003, October 2023. doi: 10.1109/ICCV51070.2023.00371. URL https://ieeexplore.ieee.org/abstract/document/10378323.

[35] Ao Sun, Pingchuan Ma, Yuanyuan Yuan, et al. Explain Any Concept: Segment Anything Meets Concept-Based Explanation. In *Advances in Neural Information Processing Systems*, volume 36, pages 21826–21840. Curran Associates, Inc., 2023. doi: 10.52202/075280-0954. URL https://dl.acm.org/doi/10.5555/3666122.3667076.

[36] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, et al. Concept Bottleneck Models. In *Proceedings of the 37th International Conference on Machine Learning*, pages 5338–5348. PMLR, November 2020. URL https://proceedings.mlr.press/v119/koh20a.html.

[37] Hasan Md Tusfiqur Alam, Devansh Srivastav, Md Abdul Kadir, et al. Towards Interpretable Radiology Report Generation via Concept Bottlenecks Using a Multi-agentic RAG. In Claudia Hauff, Craig Macdonald, Dietmar Jannach, Gabriella Kazai, Franco Maria Nardini, Fabio Pinelli, Fabrizio Silvestri, and Nicola Tonellotto, editors, *Advances in Information Retrieval*, pages 201–209, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-88714-7. doi: 10.1007/978-3-031-88714-7_18.

[38] Mor Geva, Roei Schuster, Jonathan Berant, et al. Transformer Feed-Forward Layers Are Key-Value Memories. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.446. URL https://aclanthology.org/2021.emnlp-main.446/.

[39] Nina Rimsky, Nick Gabrieli, Julian Schulz, et al. Steering Llama 2 via Contrastive Activation Addition. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.828. URL https://aclanthology.org/2024.acl-long.828/.

[40] Ole Jorgensen, Dylan Cope, Nandi Schoots, et al. Improving Activation Steering in Language Models with Mean-Centring. arXiv preprint arXiv:2312.03813, December 2023. URL http://arxiv.org/abs/2312.03813.

[41] Hyunsoo Kim and Haesun Park. Nonnegative Matrix Factorization Based on Alternating Nonnegativity Constrained Least Squares and Active Set Method. *SIAM Journal on Matrix Analysis and Applications*, 30(2):713–730, 2008. doi: 10.1137/07069239X. URL https://doi.org/10.1137/07069239X.

[42] Chris H.Q. Ding, Tao Li, and Michael I. Jordan. Convex and Semi-Nonnegative Matrix Factorizations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1):45–55, January 2010. ISSN 1939-3539. doi: 10.1109/TPAMI.2008.277. URL https://ieeexplore.ieee.org/document/4685898.

[43] Shuai Bai, Keqin Chen, Xuejing Liu, et al. Qwen2.5-VL Technical Report. arXiv preprint arXiv:2502.13923, February 2025. URL http://arxiv.org/abs/2502.13923.

[44] Jack Hessel, Ari Holtzman, Maxwell Forbes, et al. CLIPScore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7514–7528, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.595. URL https://aclanthology.org/2021.emnlp-main.595/.

[45] Tianyi Zhang, Varsha Kishore, Felix Wu, et al. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SkeHuCVFDr.

[46] Md Abdul Kadir, Amir Mosavi, and Daniel Sonntag. Evaluation Metrics for XAI: A Review, Taxonomy, and Practical Applications. In *2023 IEEE 27th International Conference on Intelligent Engineering Systems (INES)*, pages 000111–000124, July 2023. doi: 10.1109/INES59282.2023.10297629. URL https://ieeexplore.ieee.org/document/10297629.

[47] Mara Graziani, An-phi Nguyen, Laura O'Mahony, et al. Concept discovery and Dataset exploration with Singular Value Decomposition. In *ICLR 2023 Workshop on Pitfalls of Limited Data and Computation for Trustworthy ML*, 2023. URL https://openreview.net/forum?id=iOlYmD1PtC8.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: Our experiments and theory fully support the main claim and contributions stated in the abstract and introduction.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We disucuss the limitation of our work in the main paper as well as a deginated section in the appendix.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [N/A]

Justification: Our results are based on empirical analyses.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We have included all relevant information supporting reproducibility in the Supplementary material.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: The project website and code link are provided in the abstract.

Guidelines:

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

6. **Experimental setting/details**

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

Answer: [Yes]

Justification: Experiments are setup following previous baselines and benchmarks. All relevant information and hyperparameters are included in the paper and the Appendix.

Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [Yes]

Justification: Results are reported as mean $\pm$ std over evaluation images. Faithfulness comparisons include Welch's t-tests with Holm correction and Cohen's d, plus a random-order control.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We show empirical time consumption but not the theoretical analysis on it.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We follow the guidelines of NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [N/A].

    Justification: Our paper does not have known risk and social impact.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: We haven't provided any generative model. It does not apply to our method.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We cited the borrowed material properly.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: We documented the implementation details and provided the code.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: Does not include human user study.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: Does not require human study.

    Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [Yes]

Justification: LVLMs (multimodal LLMs) are used within our method and experiments: a frozen LVLM generates the candidate concept vocabulary, answers the binary concept-presence prompts, and is the model whose hidden states we decompose and explain (Section 3, Appendix C). LLMs were not used to develop the method or to generate the paper's content. See the "Broader impact" paragraph in Section H.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.

SUPPLEMENTARY MATERIAL FOR "CONCEPT LEARNING FOR INTERPRETING LARGE VISION–LANGUAGE MODELS"

Contents

1 Introduction **1**

2 Related Work **2**

3 Method **3**

3.1 Positive and Negative Concept Bags 3

3.2 Text-Guided Dictionary Learning 4

3.3 Positive Concept Vector Identification 5

3.4 Concept Attribution and Multi-modal Alignment 6

4 Experiments **6**

4.1 Experiments and Metrics 6

4.2 Results 7

5 Ablation Study **9**

6 Conclusion **10**

Supplementary Material **22**

A Preliminaries **24**

B Monosemantic vs. Polysemantic Representations **24**

C Prompts **24**

D Models **25**

D.1 Gemma-3n E4B-IT 25

D.2 Qwen2.5-VL-7B-Instruct and Qwen2-VL-7B-Instruct 25

E Dataset Descriptions **26**

F Results **26**

F.1 Qwen 2.0-VL-7B 26

F.2 Qwen 2.5-VL-7B 26

F.3 Posthoc Concept Explanations for LVLMs 27

F.4 Concept vectors are generalizable to related objects 28

F.5 Explanation with abstract concepts 28

F.6 Grounding limitation 28

G Ablation study **30**

H Computational Cost Analysis **33**

A Preliminaries

Recent work on concept vector extraction [14, 35, 19, 9] shows that many of these methods can be framed as *dictionary learning*, where activations are approximated by a small set of interpretable vectors. Formally, given activations $S \in \mathbb{R}^{n \times d}$ with n samples and d-dimensional vector, we posit K latent concepts and solve

$$\underset{U \in \mathbb{R}^{d \times K},\, V \in \mathbb{R}^{n \times K}}{\arg\min} \|S - VU^\top\|_F^2,$$

where $U = [u_1, \ldots, u_K]$ are the concept vectors (CAVs), and $V = [v_1^\top; \ldots; v_n^\top]$ are the concept activations, with the row v_i giving the concept coordinates for sample i.

This factorization unifies prior approaches via constraints on (U, V):

$$\begin{cases} v_i \in \{e_1, \ldots, e_K\}, & \text{K-Means [14],} \\ U^\top U = I, & \text{PCA [47],} \\ S \geq 0,\ U \geq 0,\ V \geq 0, & \text{NMF [19],} \\ U \text{ free},\ V \geq 0, & \text{SNMF [33],} \\ V = \psi(S),\ \|v_i\|_0 \leq s, & \text{SAE [18].} \end{cases}$$

Columns in U are concept vectors (CAVs), and rows in V are per-sample activations. Special cases include PCA (orthogonal bases), NMF (nonnegative factors), SNMF (mixed-sign basis vectors while constraining the corresponding activations to be nonnegative), K-Means (cluster centers and distance of vectors), and SAE (encoder–decoder with sparsity).

B Monosemantic vs. Polysemantic Representations

A central challenge in interpreting large vision–language models (LVLMs) lies in *superposition* and *feature entanglement* in high-dimensional residual streams [8]. Here, *features* can be understood as vector directions in activation space that encode candidate concepts. Ideally, such vectors should be *monosemantic*, each aligned with a single interpretable concept. In practice, however, LVLMs often learn *polysemantic* vectors, where multiple, semantically unrelated concepts activate a single direction.

For instance, in Figure 5(a), a feature f_1 responds to both "cat" and "chair." Such overlap can arise when these concepts frequently co-occur in training data, leading the model to conflate them. When it is activated, it is therefore ambiguous whether the cause was the presence of a cat, a chair, or both. This ambiguity breaks the one-to-one mapping between features and concepts, making attribution unreliable. In this paper, we investigate how to extract monosemantic concept vectors from the polysemantic features that neurons of a model activate during a prediction.

Monosemantic features (Figure 5(b)) provide a clean relation to concepts: e.g., a concept vector is an approximation of monosemantic features.

Apart from a limited number of studies [29, 18], most existing models do not offer disentangled representations, and tools for analyzing and extracting monosemantic features remain scarce. This paper addresses this gap through TGCL.

C Prompts

Concept generation prompt. We use the following instruction to extract candidate concept text from images: *"Identify every visible object, item, concept, and pattern in the image. Output only a single-word, comma-separated list. No explanations or sentences."* This prompt generates concept tokens directly from the dataset without requiring manual labels, thereby creating a concept-to-image mapping.

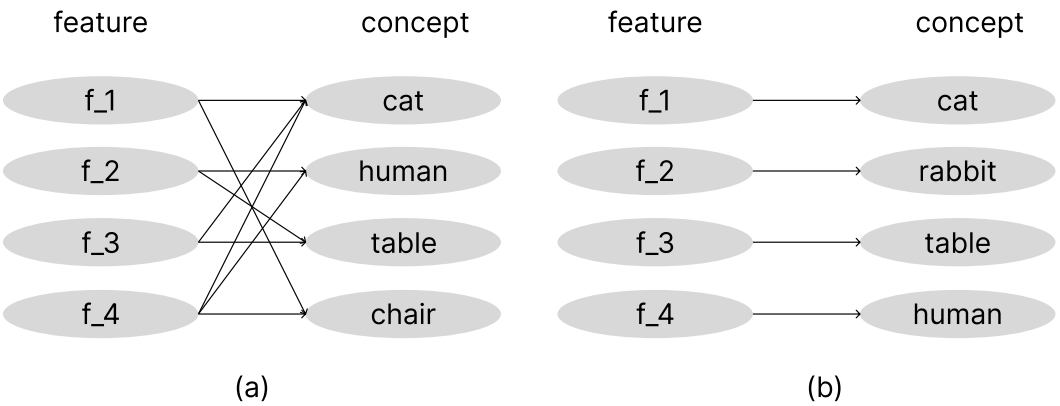


Figure 5: Comparison between prior concept decomposition methods and our proposed approach. (a) Previous methods (e.g., Parekh et al. [9]) often produce polysemantic features (e.g., f_3 activates for "chair" and "cat"). (b) Our method encourages monosemantic features (e.g., f_1 for "cat," f_3 for "table").

Text-guided prompt

- P1: Detect whether the image contains c_k. If yes, return c_k; otherwise, return `UNK`.
- P2: Does the image contain c_k? If yes, output c_k; otherwise, output No-c_k.
- P3: Is there a clear instance of c_k in this image? Reply with c_k or `thing`, nothing else.
- P4: Recognize whether the concept c_k is present in the picture. Use only c_k or `UNK` as your answer.

Table 5: Mean $\pm$ std. of CLIPScore across binary prompts on MSCOCO-10. Abbreviations: Q-2 = Qwen2.5-VL-7B, Q-2.5 = Qwen2.5-VL-7B, G-3n = Gemma-3n-E4B.

Prompt	Q-2	Q-2.5	G-3n	Prompt	Q-2	Q-2.5	G-3n
P1	0.57±0.10	0.65±0.13	0.62±0.08	P3	0.57±0.14	0.62±0.07	0.62±0.06
P2	0.58±0.07	0.63±0.11	0.64±0.08	P4	0.59±0.11	0.63±0.08	0.64±0.09

D Models

D.1 Gemma-3n E4B-IT

Gemma-3n-4B-IT [3] is a multimodal instruction-tuned model with approximately 4B effective parameters (about 8B raw parameters). It is based on the Matryoshka Transformer (MatFormer) architecture, which nests smaller subnetworks within a single model; the E4B model contains the smaller E2B subnetwork. The language model uses 35 Transformer layers, a hidden size of 2,048, 8 query heads, and 2 key-value heads, with sliding-window attention (window size 512) in most layers and full attention in every fifth layer. It further uses Per-Layer Embeddings (PLE), which reduce the accelerator memory required per parameter, and shares key-value caches across the last 15 layers. It accepts text, audio inputs, and images, with a MobileNet-V5 vision encoder, and supports a context length of 32,768 tokens. We use the instruction-tuned E4B model with all parameters frozen.

D.2 Qwen2.5-VL-7B-Instruct and Qwen2-VL-7B-Instruct

Qwen2.5-VL-7B-Instruct [43] is a 7B-class instruction-tuned vision–language model that accepts text and images (as well as videos) as input and generates text. It combines a Qwen2.5 language model with a native dynamic-resolution vision transformer (ViT) trained from scratch. The language model consists of 28 transformer layers with a hidden size of 3,584. The ViT contains 32 transformer layers, with four layers using full attention and the remaining layers using window attention. Multimodal

rotary positional embeddings (M-RoPE) are used to encode positional information across text, images, and videos. The model was pretrained on 4.1T tokens, with sequences of up to 32,768 tokens in the final pretraining stage, and subsequently underwent supervised fine-tuning and direct preference optimization. We use the model with all parameters frozen.

Qwen2-VL-7B-Instruct [2] is the predecessor of Qwen2.5-VL. It combines a Qwen2-based language model with a vision transformer of approximately 675M parameters and employs Naive Dynamic Resolution, allowing images to be represented using a variable number of visual tokens according to their resolution. The language model consists of 28 transformer layers with a hidden size of 3,584. The model uses Multimodal Rotary Position Embedding (M-RoPE) to encode positional information across text, images, and videos. It supports a context length of up to 32,768 tokens and was subsequently instruction-tuned for multimodal generation. We use the model with all parameters frozen.

E Dataset Descriptions

We evaluate our models on four datasets: **ImageNet** [10], **MSCOCO** [11], **CIFAR100** [12], and **DTD (Describable Textures Dataset)** [13].

- **ImageNet**: A large-scale visual dataset with 1000 object categories and high-resolution images, commonly used for image classification tasks.
- **MSCOCO**: A subset of the MSCOCO dataset with 10 object categories, featuring complex scenes and multiple annotated objects per image.

For each dataset, after constructing the concept vocabulary, we select the top 1000 most frequent concepts from ImageNet, the top 10 from MSCOCO, the top 100 from CIFAR100, and the top 47 from DTD. These numbers are chosen to align the size of the concept vocabulary with the number of ground-truth classes in each dataset, ensuring comparable concept coverage across datasets.

Dataset	Total concepts	Train Samples/Class	Val Samples/Class
ImageNet	1000	300	50
MSCOCO (10)	10	300	50
CIFAR100	100	300	100
DTD	47	95	24

Table 6: Overview of datasets used for training and validation, including the number of classes and samples per class.

F Results

F.1 Qwen 2.0-VL-7B

Table 7 reports the performance of **TGCL** and **CoX-LMM** on four benchmark datasets. We evaluate alignment using **CLIPScore (CS)** and **BERTScore (BS)**, where higher is better.

Across all datasets, TGCL consistently outperforms CoX-LMM. Notably, on **MSCOCO**, TGCL achieves the strongest gains: BS improves from 0.82 (CoX-LMM) to **0.94**, and CS improves from 0.58 to **0.64**. Similarly, on **CIFAR100**, the BS of CoX-LMM drops to 0.73 in the combined setting, while TGCL achieves a significantly higher **0.91**. These results highlight the robustness of TGCL in both low- and high-concept regimes.

F.2 Qwen 2.5-VL-7B

Table 8 shows results for the Qwen2.5 backbone. Again, TGCL achieves improvements across all datasets. In particular, on **MSCOCO**, TGCL improves BS from 0.90 to **0.95** in the combined setting. On **CIFAR100**, TGCL reaches **0.92**, compared to 0.87 with CoX-LMM. These consistent gains indicate that TGCL scales effectively from small (10 concepts) to large-scale (1k concepts) benchmarks.

Table 7: Comparison of **TGCL** and **CoX-LMM** across datasets. Metrics: CS = CLIPScore, BS = BERTScore. Higher is better. The best non-random results are in bold.

Method	Dataset	#C	Metric	Rand	Text	Img	Comb
TGCL	ImageNet	1k	CS	0.53 ± 0.02	–	**0.62 ± 0.07**	**0.63 ± 0.08**
			BS	0.80 ± 0.05	**0.86 ± 0.06**	**0.88 ± 0.08**	**0.86 ± 0.09**
	CIFAR100	100	CS	0.51 ± 0.02	–	**0.63 ± 0.04**	**0.63 ± 0.05**
			BS	0.83 ± 0.08	**0.86 ± 0.07**	**0.87 ± 0.08**	**0.91 ± 0.08**
	DTD	47	CS	0.53 ± 0.06	–	**0.63 ± 0.05**	**0.62 ± 0.05**
			BS	0.74 ± 0.04	**0.84 ± 0.06**	**0.83 ± 0.07**	**0.87 ± 0.06**
	MSCOCO	10	CS	0.52 ± 0.03	–	**0.64 ± 0.06**	**0.62 ± 0.06**
			BS	0.82 ± 0.02	**0.88 ± 0.06**	**0.89 ± 0.05**	**0.94 ± 0.08**
CoX-LMM	ImageNet	1k	CS	0.53 ± 0.03	–	0.54 ± 0.03	0.53 ± 0.05
			BS	0.82 ± 0.04	0.82 ± 0.03	0.84 ± 0.04	0.86 ± 0.09
	CIFAR100	100	CS	0.53 ± 0.04	–	0.53 ± 0.06	0.53 ± 0.05
			BS	0.78 ± 0.06	0.84 ± 0.06	0.80 ± 0.03	0.73 ± 0.07
	DTD	47	CS	0.51 ± 0.05	–	0.54 ± 0.05	0.52 ± 0.05
			BS	0.80 ± 0.07	0.84 ± 0.06	0.83 ± 0.07	0.77 ± 0.07
	MSCOCO	10	CS	0.53 ± 0.03	–	0.57 ± 0.04	0.58 ± 0.06
			BS	0.82 ± 0.04	0.83 ± 0.01	0.83 ± 0.05	0.82 ± 0.03

Method	Dataset	#C	Metric	Rand	Text	Img	Comb
TGCL	ImageNet	1k	CS	0.51 ± 0.06	–	**0.63 ± 0.06**	**0.64 ± 0.05**
			BS	0.82 ± 0.04	**0.87 ± 0.01**	**0.88 ± 0.07**	**0.87 ± 0.07**
	MSCOCO	10	CS	0.53 ± 0.05	–	**0.64 ± 0.07**	**0.64 ± 0.08**
			BS	0.83 ± 0.06	**0.89 ± 0.07**	**0.90 ± 0.06**	**0.95 ± 0.08**
	CIFAR100	100	CS	0.50 ± 0.07	–	**0.62 ± 0.05**	**0.64 ± 0.06**
			BS	0.80 ± 0.08	**0.88 ± 0.06**	**0.88 ± 0.09**	**0.92 ± 0.07**
	DTD	47	CS	0.51 ± 0.06	–	**0.63 ± 0.08**	**0.63 ± 0.07**
			BS	0.79 ± 0.07	**0.87 ± 0.07**	**0.85 ± 0.09**	**0.89 ± 0.08**
CoX-LMM	ImageNet	1k	CS	0.51 ± 0.05	–	0.56 ± 0.04	0.55 ± 0.06
			BS	0.81 ± 0.06	0.83 ± 0.04	0.85 ± 0.06	0.86 ± 0.03
	CIFAR100	100	CS	0.53 ± 0.07	–	0.58 ± 0.06	0.57 ± 0.04
			BS	0.78 ± 0.07	0.84 ± 0.06	0.85 ± 0.05	0.87 ± 0.06
	MSCOCO	10	CS	0.52 ± 0.03	–	0.60 ± 0.03	0.60 ± 0.02
			BS	0.81 ± 0.04	0.88 ± 0.04	0.90 ± 0.04	0.90 ± 0.07
	DTD	47	CS	0.53 ± 0.03	–	0.56 ± 0.03	0.56 ± 0.06
			BS	0.81 ± 0.04	0.83 ± 0.05	0.82 ± 0.06	0.88 ± 0.05

Table 8: Comparison of TGCL and CoX-LMM across datasets using CLIPScore (CS) and BERTScore (BS). Best non-random results are in bold.

F.3 Posthoc Concept Explanations for LVLMs

Unlike the existing methods, which can't provide any token-level post-hoc explanation, our method provides token-level explanations in autoregressive Large Vision-Language Models (LVLMs). We present qualitative examples in Figures 8 6 and 9 for Qwen2.5-VL-7B. Figure 6 results belong to the experiment using SAM as a localizer mentioned in 3.1, while Figure 8 and 9 present the concept attribution example using a random cropping localizer during concept. Each example uses

a structured 2×2 image grid that intentionally makes the prediction task more challenging while encouraging the model to produce structured, multi-object descriptions.

On the **left** of each example, we show the *input image* (the 2×2 grid) together with the *prompt* used for generation. Directly below, we display the *LVLM's textual output* produced for this visual input.

On the **right**, we visualize token-wise concept activations. For each generated token, we extract its penultimate-layer embedding and compute its cosine similarity to all learned concept vectors. For token-to-word mapping, we consider only the first produced token embedding for that position when calculating the cosine distance to the concept vectors. We then identify the top-2 most activated concepts (from left to right) corresponding to that token.

The right-hand panel contains:

- a *concept grid* showing, for each token, the two highest-scoring concepts;
- a *concept bank* in the form of a row of thumbnails, starting with a similarity bar that shows the cosine similarity between the token embedding and the corresponding concept vector;
- a short *textual grounding label* above each concept bank, summarizing the semantic meaning of the discovered concept.

We find that SAM and the random localizer perform similarly in attribution ranking: both methods correctly attribute the concept images. The main difference is that SAM yields better visualizations for concepts such as hot-dog, beaver, and bear in the concept bank. Moreover, in Figure 7, we show qualitative examples of explanations produced by the baseline CoX-LMM to highlight its limitations and how our method improves the ranking of similar concepts. While the CoX-LMM concept vector often shows high cosine similarity with the token activations, the corresponding concept bank contains many different object types. This indicates that multiple concept signals are entangled in a single vector, i.e., the concept vectors are polysemantic. Such explanations are less useful because we cannot reliably map the model output to a specific, human-interpretable concept. In addition, some textual concepts associated with the concept bank are not clearly related to the underlying visual patterns.

F.4 Concept vectors are generalizable to related objects

To study the robustness and generalization of our concept-based explanations, we also analyze cases where the *object-specific concept is missing in the concept dictionary*. In Figure 9—10, we provide qualitative examples showing how the LVLM aligns its predictions with the *closest available* concepts, even when the extracted concept set belongs to different object classes than the input images.

As in the previous section, F.3, each example contains a structured 2×2 input grid. On the left side of each example, we show the input image, the prompt, and the LVLM's generated output. Here, the learned concepts originate from *different object categories* than the input images. Despite this mismatch, the LVLM often activates semantically *related* concepts whose attributes partially overlap with the visual content (e.g., "stripe-like patterns", " texture", "fur-like appearance").

F.5 Explanation with abstract concepts

We explored how abstract concepts relate to an LVLM's outputs using ImageNet validation examples. In Figure 11, we present three cases. We found that our explanation method captures clear relationships between predicted tokens and related abstract concepts. For instance, *macaw* shows high similarity to *colorful* concepts; *ladybug* and *spotted dog* (Dalmatian) show high similarity to *polka-dot*; and *jellyfish* shows high similarity to *skin* and *soft* concepts. These results shed light on how the model may internally perform abstract reasoning when predicting a token.

F.6 Grounding limitation

Although our concept-based attribution method generally provides coherent visual and textual grounding, we observe an important failure case when analyzing images containing a *beaver* property. Figure 12 illustrates this phenomenon.

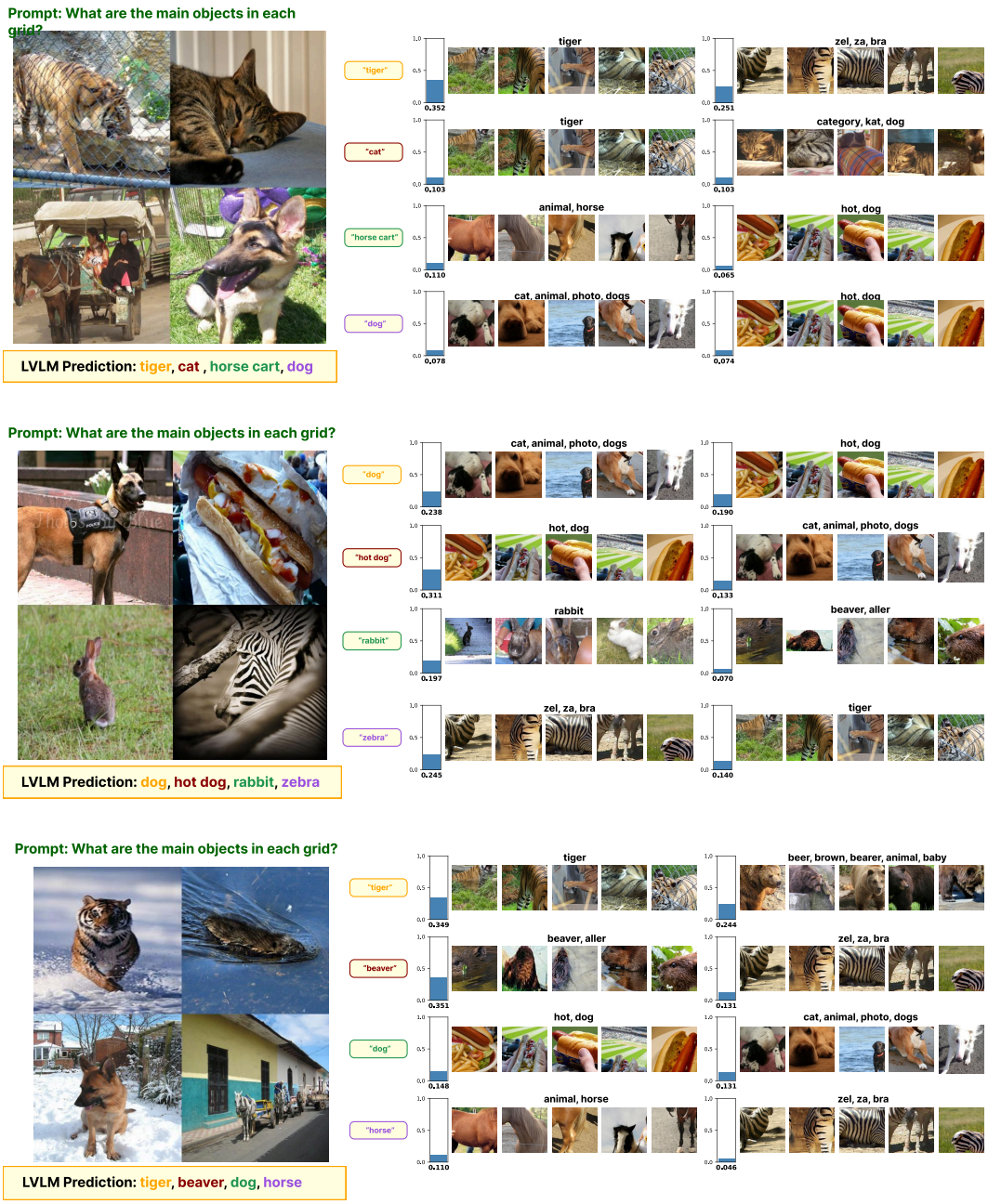


Figure 6: **Left: input 2×2 grid, prompt, and LVLM output. Right: token-wise top-2 concept activations, cosine-similarity bars, and textual grounding and visual grounding. Example using SAM as an object localizer during concept extraction.**

Shifted textual grounding in some examples. *Textual grounding* is sometimes slightly shifted (Figure 12) or offset from the intended semantic meaning (e.g., the "cat" concept is grounded as "dog," "category," and "kat," and for "beaver," it shifted to the tokens "based" and "prediction"), while the visual grounding is consistent with the image of a "cat" and "beaver."

This discrepancy suggests that while the concept vectors are visually stable and reliably activated across different images, the mapping from concept vectors to text tokens remains sensitive to local variations in the LVLM's decoder distribution. As a result, text grounding may deviate slightly even when image grounding is fully correct. Overall, however, the image-side concept activations in our concept-based examples remain remarkably uniform and consistent across all inputs.

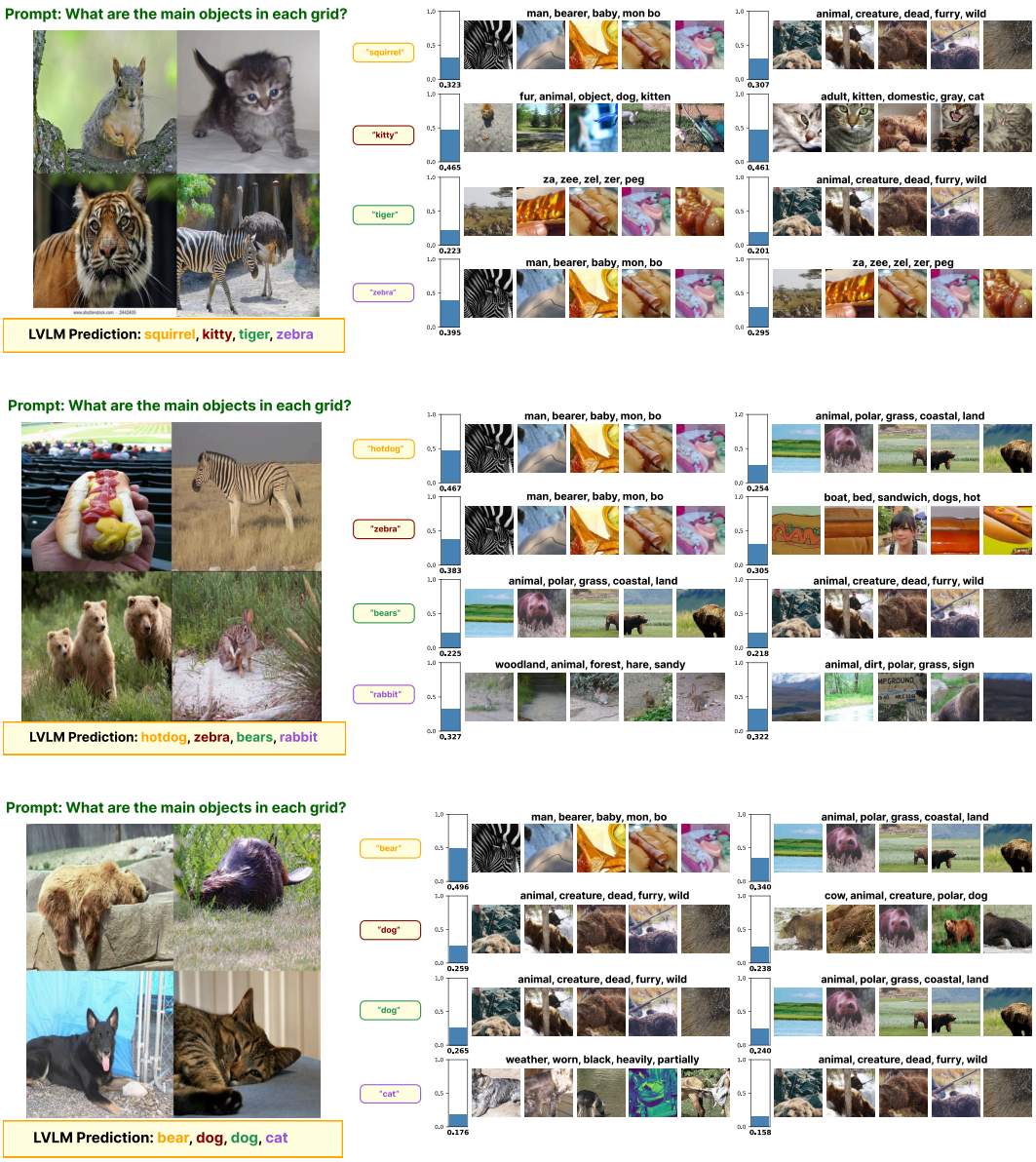


Figure 7: Left: input 2×2 image grid, prompt, and LVLM output. Right: token-wise top-2 concept activations, cosine-similarity bars, and textual/visual grounding obtained with the baseline CoX-LMM. SAM is used as an object localizer during concept extraction.

G Ablation study

Dictionary size ablation on Gemma-3n-E4B. We perform an ablation on the dictionary size K (number of atoms) (Table 9) using Gemma-3n. Dictionaries are learned on ImageNet training data and evaluated on the ImageNet validation split from the same five object classes (chosen to reduce computation). For each K, we measure CLIPScore for image–text alignment and BERTScore for semantic similarity of the attributed concept and prediction.

SNMF α ablation on Gemma-3n-E4B. We also ablate the SNMF sparsity weight α while keeping the dictionary size fixed. Larger α promotes sparser and more selective atoms, while smaller α yields denser activations. We train on ImageNet training images from the same five classes and evaluate on the corresponding validation split, reporting CLIPScore and BERTScore for each α. As shown in

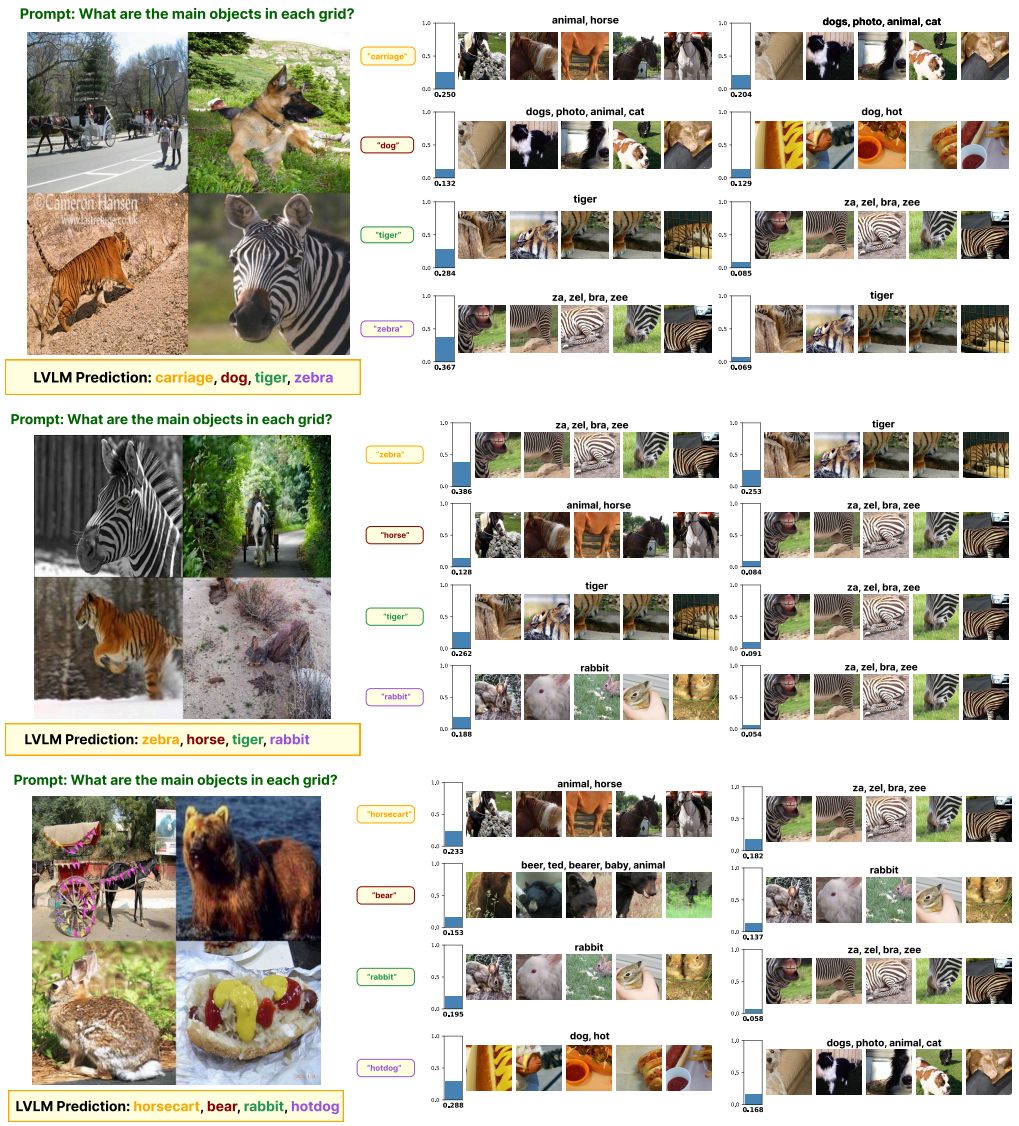


Figure 8: Example using random cropping as an object localizer during concept extraction using TGCL.

Table 10, increasing α gives only a slight improvement, and the overall differences are small. Based on this study, we use $\alpha = 20$ in all main experiments.

SAM vs. non-SAM localization ablation. We further ablate the image localization step by comparing SAM-based region proposals with a non-SAM baseline (random/local crops), while keeping the LVLM (Gemma-3n-E4B) and dictionary settings fixed. Both variants are trained on ImageNet training images and evaluated on the validation split of the same five classes. We report CLIPScore and BERTScore to quantify concept quality. As shown in Table 11, the two approaches achieve similar quantitative performance. However, Figure 6 and Figure 8 show that SAM produces cleaner and more visually coherent concept exemplars for some classes than the random localizer, improving qualitative interpretability.

Layer Ablation We extract concept vectors from different normalization layers of Gemma-3n-E4B and report their BERTScore and CLIPScore in Figure 13. We observe that the CLIP scores follow the same trend as in prior work: they are generally higher for deeper layers [9]. This is expected,



Figure 9: **More examples using random localizer**

since deeper layers contain more global image features, and CLIPScore primarily measures global image–text similarity rather than fine-grained local details.

In contrast, the BERT scores do not increase monotonically with depth. Instead, they are high for some layers and low for others. This is reasonable because BERTScore only compares the text descriptions of the concepts. Text is discrete and does not decompose into "low-level" vs. "high-level" visual features in the same way as image representations, so BERTScore is largely agnostic to layer depth. A high BERTScore for a layer indicates that its concept vectors yield coherent and semantically rich textual concepts, whereas a low BERTScore suggests that the corresponding layer provides a poorer representation of the underlying concepts.

Concept Bag size Ablation We ablate TGCL under different concept bag sizes. Larger bags provide more concept examples via random cropping. We train on ImageNet samples from five classes and evaluate on the corresponding validation split, reporting CLIPScore and BERTScore on



Figure 10: **Concept-Mismatch Analysis.** When the exact object's concept does not exist in the dictionary, the LVLM aligns its token embeddings with the most semantically related available concepts.

the combined prediction task for varying bag sizes. As shown in Table 12, increasing the bag size improves both CLIP and BERT scores, suggesting that larger bag sizes are beneficial, although the gains begin to saturate around a bag size of 900.

H Computational Cost Analysis

Setup. We compare the computational cost of **TGCL** and the **CoX-LMM** (SOTA) under the same hardware: GPU: **NVIDIA RTX 3090 (24GB)**, CPU: **AMD Ryzen 9 5950X (16 cores)**. On this setup, extracting concepts for a fixed set of 10 objects takes **421 s** for TGCL and **4375 s** for CoX-LMM

Summary. Table 13 reports the estimated training and inference costs. TGCL requires substantially fewer FLOPs than CoX-LMM during concept learning, with approximately a 40% reduction in GPU

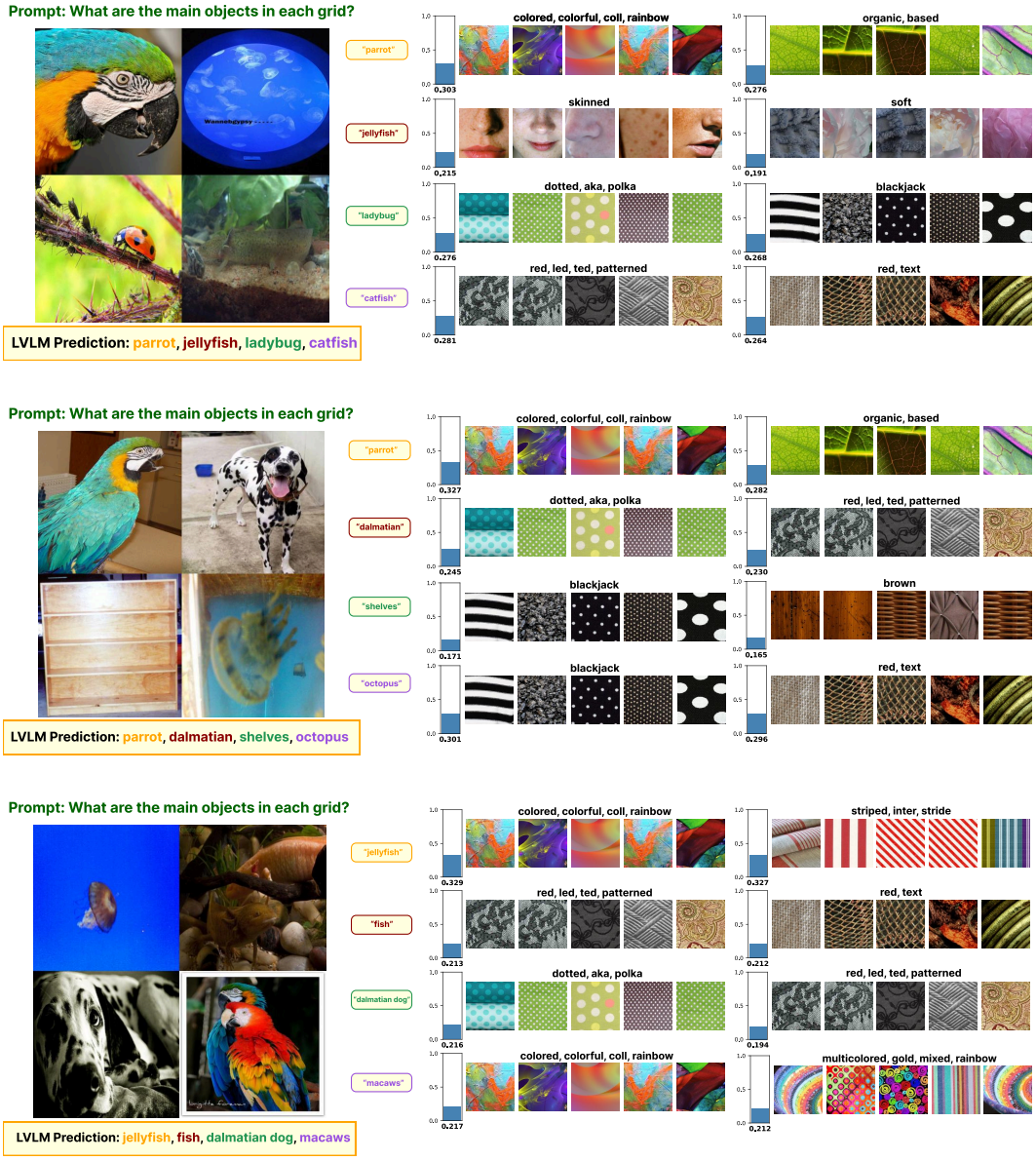


Figure 11: **Examples:** Top-2 concept activations reveal partial semantic similarity between token prediction and related abstract concepts.

FLOPs and approximately 56% higher CPU operations. Empirical results further suggest that TGCL is about $10\times$ faster than CoX-LMM, indicating that GPU FLOPs are the primary computational bottleneck.

Decomposition of TGCL FLOPs. We approximate the LVLM cost using a Gemma-3n-4B backbone with $M = 4 \times 10^9$ parameters and assume a per-token cost of $\approx 2M \approx 8 \times 10^9$ FLOPs. Let T denote the total token length of the multimodal sequence (image tokens, prompt tokens, and generated tokens).

(1) Concept-bag creation (Sec. 4.1). We create concept bags from $N_{\text{img}} = 3000$ images (300 images per category) with token length $T = 196 + 40 + 200 = 436$ per sample. The total cost for this stage is

$$\text{FLOPs}_{\text{imgs}} \approx (8 \times 10^9) \cdot T \cdot N_{\text{img}} \approx 1.05 \times 10^{16}.$$

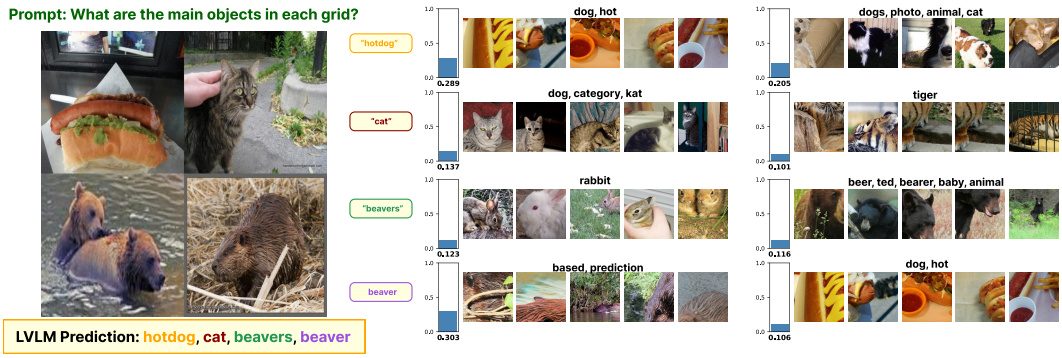


Figure 12: **Failure Case: LVLM Hallucination, Text Grounding Incorrect in TGCL**. LVLM predicts "bears'" image as beavers. The concept explanation relates more to a rabbit than a bear. This means the model knows bears, but it hallucinated because of the beaver-like appearance. Secondly, even though the textual grounding for "beaver" is incorrect in the last concept bank in the middle, image grounding is correct, and activation correctly responds to the beaver-like region (body on water). The textual grounding of the concept is incorrectly shifted due to interference (possibly "water-based"), demonstrating a misalignment between visual and textual grounding.

K	BERT@1 ↑	BERT@2 ↑	BERT@3 ↑	CLIP@1 ↑	CLIP@2 ↑	CLIP@3 ↑
2	0.881±0.013	0.884±0.014	0.885±0.014	0.616±0.044	0.638±0.041	0.652±0.043
10	0.881±0.013	0.884±0.013	0.885±0.013	0.603±0.047	0.629±0.047	0.643±0.048
30	0.884±0.040	0.891±0.041	0.892±0.041	0.475±0.033	0.513±0.026	0.532±0.026
50	0.885±0.040	0.891±0.041	0.892±0.041	0.509±0.028	0.532±0.028	0.552±0.030
100	0.887±0.041	0.891±0.041	0.892±0.041	0.488±0.035	0.512±0.031	0.525±0.032

Table 9: Ablation over the number of concept atoms K in the dictionary (five sampled settings). BERTScore stays roughly constant around 0.88, while CLIPScore generally decreases as K increases. Values are mean±std over five ImageNet classes.

Table 10: SNMF sparsity weight α ablation on Gemma-3n (5 ImageNet classes). Values are mean ± std.

α	BERTScore ↑			CLIPScore ↑		
	@1	@2	@3	@1	@2	@3
0	0.881±0.013	0.884±0.013	0.885±0.013	0.610±0.039	0.634±0.036	0.646±0.037
20	0.881±0.013	0.884±0.014	0.885±0.014	0.616±0.044	0.638±0.041	0.652±0.043
100	0.881±0.013	0.884±0.014	0.885±0.014	0.615±0.043	0.638±0.040	0.652±0.042
150	0.881±0.013	0.884±0.014	0.885±0.013	0.617±0.043	0.639±0.038	0.653±0.039
200	0.881±0.013	0.883±0.013	0.884±0.014	0.621±0.039	0.644±0.039	0.659±0.042

(2) Segmentation cost (SAM) (Sec. 4.1). From these images, we obtain 16,000 object-centric crops using SAM. Approximating the SAM forward cost as $\approx 1.1 \times 10^{12}$ FLOPs per image, we obtain

$$\text{FLOPs}_{\text{SAM}} \approx (1.1 \times 10^{12}) \cdot 3000 \approx 3.3 \times 10^{15}.$$

(3) Residual-stream extraction for concept bags (Sec. 4.2). For dictionary learning we use $N_{\text{res}} = 16{,}000$ residual samples (1600 crops per concept, 10 concepts). During binary prompting, the token length is $T_{\text{res}} \approx 196 + 21 + 10 = 227$. The LVLM FLOPs for residual extraction are

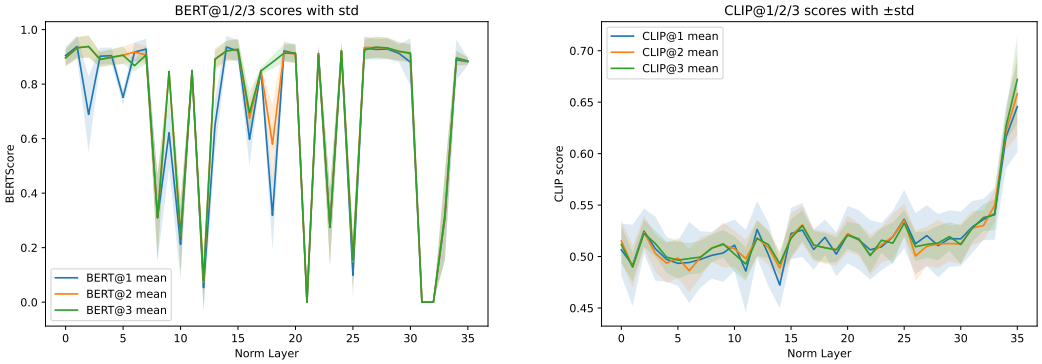
$$\text{FLOPs}_{\text{res}} \approx (8 \times 10^9) \cdot T_{\text{res}} \cdot N_{\text{res}} \approx 2.91 \times 10^{16}.$$

(4) SNMF dictionary learning on CPU (Sec. 4.2). We factorize the residual activations with sparse NMF using $N = 1600$ samples, feature dimension $D = 2048$, dictionary size $K = 2, 5000$ iterations, and 10 concepts:

$$\text{Ops}_{\text{TGCL,SNMF}} \approx N \cdot D \cdot K \cdot \text{iters} \cdot \text{Concepts} = 1600 \cdot 2048 \cdot 2 \cdot 5000 \cdot 10 \approx 3.28 \times 10^{11} \text{ CPU ops.}$$

Model	SAM		Random	
	BERTScore@1 ↑	CLIPScore@1 ↑	BERTScore@1 ↑	CLIPScore@1 ↑
Gemma-3n	0.93 ± 0.04	0.61 ± 0.02	0.88 ± 0.01	0.59 ± 0.04
Qwen-2.5	0.92 ± 0.03	0.67 ± 0.03	0.91 ± 0.06	0.66 ± 0.03
Qwen-2.0	0.93 ± 0.04	0.63 ± 0.04	0.91 ± 0.06	0.61 ± 0.04

Table 11: SAM vs. Random localization ablation across three LVLM backbones. Values are mean $\pm$ std over five ImageNet classes. We observe that SAM improves both CLIP and BERT over random cropping for localization.



(a) BERTScore across norm layers.

(b) CLIPScore across norm layers.

Figure 13: Layer-wise ablation on Gemma-3n. (a) BERTScore of concept phrases extracted from different norm layers. (b) CLIPScore between visual concepts and their text descriptions.

α	BERTScore ↑			CLIPScore ↑		
	@1	@2	@3	@1	@2	@3
100	0.762	0.798	0.821	0.561	0.579	0.604
200	0.774	0.811	0.829	0.578	0.591	0.613
300	0.803	0.826	0.847	0.602	0.615	0.628
400	0.831	0.848	0.862	0.619	0.629	0.641
500	0.852	0.864	0.875	0.625	0.638	0.649
700	0.867	0.876	0.883	0.631	0.643	0.654
900	0.871	0.879	0.886	0.634	0.647	0.658

Table 12: SNMF concept bag size ablation on Gemma-3n (5 ImageNet classes).

Table 13: Training and inference cost of TGCL vs. CoX-LMM.

Category	TGCL	CoX-LMM	Comment
Training GPU FLOPs	$\approx 4.28 \times 10^{16}$	$\approx 7.20 \times 10^{16}$	TGCL $\approx 40\%$ cheaper
Training CPU ops (dictionary learning)	$\approx 3.28 \times 10^{11}$	$\approx 1.43 \times 10^{11}$	CoX-LMM $\approx 56\%$ lower
Inference FLOPs/image	$\approx 1.6 \times 10^{11}$	$\approx 1.6 \times 10^{11}$	Cosine cost negligible

Total TGCL cost. Summing the LVLM FLOPs across stages yields

$$\text{FLOPs}_{\text{TGCL,total}} \approx 1.05 \times 10^{16} + 3.3 \times 10^{15} + 2.91 \times 10^{16} \approx 4.28 \times 10^{16}.$$

Decomposition of CoX-LMM FLOPs.

(1) Caption-level search (CPU). CoX-LMM first searches the MSCOCO captions to find images containing the target objects. This involves scanning $\approx$ 120,000 captions, each of length $\approx$ 400 characters, which leads to

$$\text{Ops}_{\text{CoX,search}} \approx 120{,}000 \times 400 = 4.8 \times 10^7 \text{ character comparisons.}$$

This cost is negligible compared to the LVLM forward passes.

(2) Residual-stream extraction. CoX-LMM extracts residual activations from $35{,}000$ images (≈ 3500 images per object for 10 objects). The token length is $T_{\text{res}} \approx 196 + 50 + 11 = 257$, leading to

$$\text{FLOPs}_{\text{CoX,res}} \approx (8 \times 10^9) \cdot T_{\text{res}} \cdot 35{,}000 \approx 7.20 \times 10^{16}.$$

(3) Dictionary learning on CPU. CoX-LMM uses SNMF with $N = 35{,}000$ samples, $D = 2048$, $K = 10$ concepts, and 200 iterations:

$$\text{Ops}_{\text{CoX,SNMF}} \approx N \cdot D \cdot K \cdot \text{iters} = 35{,}000 \cdot 2048 \cdot 10 \cdot 200 \approx 1.43 \times 10^{11} \text{ CPU ops.}$$

Total CoX-LMM cost. The dominant cost for CoX-LMM is the LVLM residual extraction:

$$\text{FLOPs}_{\text{CoX,total}} \approx 7.20 \times 10^{16}.$$

The additional CPU cost from caption search and SNMF is small compared to the GPU FLOPs.

Inference-time cost. At inference time, both TGCL and CoX-LMM reuse the same LVLM backbone. For a single image, we approximate the LVLM cost as $\approx 1.6 \times 10^{11}$ FLOPs, and the extra cosine similarity operations between residual activations and concept vectors are negligible. Therefore, the per-image inference cost is effectively identical for both methods.

Broader Impact

TGCL supports auditing of LVLMs by revealing the concepts they represent. Since the LVLM proposes the concepts, they may reflect its biases or hallucinations and should be read as a description of the model, not as ground truth; we see no direct risks beyond those of standard interpretability methods. **LLM use.** LLM-based vision-language models are a component of our method: a frozen LVLM generates the concept vocabulary, answers the binary presence prompts, and is the model being explained (Section 3, Appendix C). LLMs were not used to develop the method itself.