CORGI: CONCEPT-ORIENTED REDUNDANCY-GUIDED IMAGE TOKEN COMPRESSION FOR VISION-LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Visual token compression aims to reduce the computational cost of Vision-Language Models (VLMs) by removing redundant or non-informative visual tokens before or during multimodal processing. Existing visual token pruning methods suffer from two major limitations. First, their pruning decisions are semantically opaque: practitioners cannot determine what information the kept or discarded tokens represent or control generation by selecting which visual tokens should be preserved for a particular task or application. Second, many methods require layer-specific design choices and hyperparameter tuning, such as selecting the transformer layers at which pruning should be performed. We introduce the first interpretable and controllable visual token pruning framework for VLMs. Our method identifies the semantic meaning of visual tokens in natural language by leveraging the model's shared vision-language representation space. This allows an easy, human-friendly way of selecting visual tokens for pruning according to their textual meaning. The resulting token selection is performed before the visual tokens are provided to the VLM, eliminating the need for additional layer-wise token dropping or pruning schedules. Experiments on image and video benchmarks across both natural and medical domains demonstrate that our method often outperforms the state-of-the-art while offering greater interpretability, explicit control, and a simpler pruning pipeline.

1 INTRODUCTION

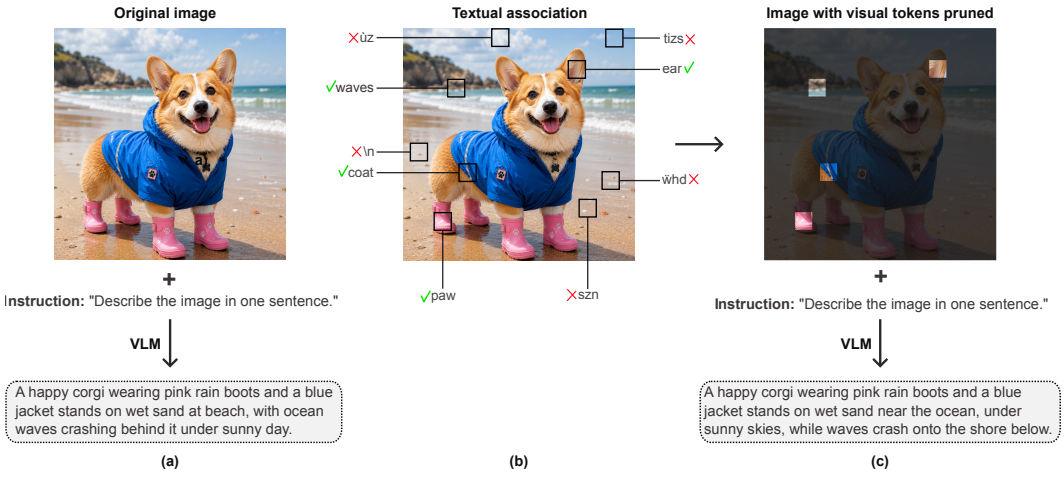


Figure 1: Illustration of our approach. (a) Baseline VLM inference processes the full set of visual tokens. (b) CORGI projects each visual patch token into the vocabulary space of the VLM to extract its associated textual meaning and filters out patches whose predicted token carries no semantic content. (c) passing only semantically grounded patches to the VLM.

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

Autoregressive Large Vision-Language Models (LVLMs) such as Qwen (Bai et al., 2025; Yang et al., 2025a) have substantially advanced multimodal perception and reasoning. Most LVLMs represent an image as a sequence of patch-level visual tokens that are projected into the language model and processed together with the textual prompt. A single image can contribute hundreds or even thousands of visual tokens as in the case of high-resolution images (e.g., medical ones) and long videos. Because the cost of self-attention increases quadratically with sequence length, processing every visual token across all transformer layers leads to substantial memory and computational consumption. Visual token pruning offers a promising direction for improving LVLM efficiency by identifying and removing meaningless and redundant tokens while retaining those that carry essential information. The central challenge is to achieve this reduction without sacrificing too much performance.

Existing approaches (Wu et al., 2026; Arif et al., 2025; Shang et al., 2025; Zhang et al., 2024; Xing et al., 2024; Yang et al., 2025b; Singh et al., 2026) suffer from two major limitations. First, their pruning decisions are difficult to interpret and semantically opaque. Tokens are typically pruned using heuristics such as attention scores, leaving users with little insight into the semantic information represented by the retained or discarded tokens. At most, such heuristics indicate that a token is likely to be important according to a particular scoring rule. Moreover, the relationship between these heuristics and true token importance is often poorly established. For example, several studies have shown that attention scores, a widely used heuristic in token pruning methods, do not reliably measure token meaningfulness (Kobayashi et al., 2020; Abnar & Zuidema, 2020), a finding that has also been confirmed specifically in the context of token pruning for large language models (LLMs) (Guo et al., 2024). This follows that existing approaches do not support controllable generation based on the information that is most relevant to a user or a specific application. In contrast, humans can describe the same image in different ways. Such control is essential in many real-world applications. For example, an autonomous driving system should prioritize potentially hazardous objects and situations in order to alert the driver, rather than describing less relevant elements such as trees or routinely observed vehicles when a risk is present. These systems should also be explainable, allowing their decisions and behavior to be interpreted, particularly when failures occur. Secondly, existing approaches rely on complex layer-wise pruning schedules and multiple hyperparameters, which compromises simplicity and practicality, and limits their transferability across different VLM architectures as these carefully tuned routines may not generalize to other models.

To address these limitations, we propose CORGI (Concept-Oriented Redundancy-Guided Image Token Compression), as illustrated in Figure 1. By leveraging the model's inherent shared vision-language representation space, CORGI projects each visual token in the vocabulary space to obtain their associated natural language semantic concept, such as *ear*, *paw*, *coat*, or *waves* (Figure 1b). This representation enables us to interpret what we select or discard from visual tokens, in a human-friendly manner, while maintaining control over the generation process. For example, we discard visual tokens associated with non-semantic vocabulary entries, including punctuation marks, formatting tokens, and unknown or malformed tokens. To achieve a higher pruning ratio, we further discard visual tokens belonging to the same semantic concept cluster. After pruning, with explicit knowledge of which tokens were removed, the remaining, kept tokens are provided to the VLM. Despite pruning nearly 90% of the visual tokens, CORGI produces a description that is close to that of the full-token baseline.

In summary, our contributions are threefold: (i) To the best of our knowledge, we introduce the first interpretable and controllable visual token pruning framework for LVLMs; (ii) our method often achieves state-of-the-art performance while avoiding the complex hyperparameter tuning and layer-wise pruning schedules required by prior approaches; and (iii) we conduct extensive experiments across a broad range of applications and domains, including natural images, videos, and medical imaging, together with comprehensive ablation studies demonstrating that each component of our method contributes non-trivially.

2 Related Work

Recent work reduces visual-token computation either before the language model or progressively within its layers. FastV (Chen et al., 2024), one of the first works that applied visual token pruning in Large Vision-Language Models, ranks visual tokens using attention scores at an early transformer

layer and prunes low-attention tokens, allowing subsequent layers to operate on a reduced visual-token set. LLaVA-PruMerge (Shang et al., 2025) selects salient visual tokens as outliers in the vision encoder's class-to-patch attention distribution, then clusters discarded tokens by key similarity and merges them into the selected representatives. HiRED (Arif et al., 2025) targets high-resolution VLMs by distributing a fixed token budget across the global image and its local partitions using early-layer CLS attention, after which final-layer CLS attention identifies the tokens retained from each partition. VisionZip (Yang et al., 2025b) similarly operates before the LLM, retaining attention-dominant tokens from the vision encoder while merging the remaining tokens into contextual representations according to feature similarity. SparseVLM (Zhang et al., 2024) instead performs text-aware pruning inside the language backbone: prompt tokens that are visually relevant act as raters for visual-token importance, while rank-based layer-wise sparsification and token recycling adapt the number of removed tokens and preserve information from discarded ones. Other approaches explicitly exploit the evolution of visual redundancy across LLM layers. PyramidDrop (Xing et al., 2024) partitions the language backbone into multiple stages and progressively removes a predefined fraction of visual tokens at each stage boundary using instruction-to-image attention. HiDrop (Wu et al., 2026) further distinguishes shallow, middle, and deep layers: it delays visual-token injection until multimodal fusion begins, applies concave progressive pruning at layers selected through inter-layer attention similarity, and removes the remaining visual tokens through an early exit. Duet-VLM (Singh et al., 2026), the recent state-of-the-art method, combines both families through vision-side dominant-token selection and local residual clustering, followed by salient-text-guided visual-token dropping at selected language-model stages.

Despite their efficiency, PruMerge, HiRED, VisionZip, and SparseVLM base selection on attention, similarity, or rank scores whose retained content remains semantically opaque and cannot be directly controlled by practitioners. PyramidDrop and SparseVLM depend on layer-wise pruning ratios, while HiDrop requires model-specific injection, filtering, and exit layers, and Duet-VLM compounds clustering choices with a multi-stage pruning schedule. In contrast, CORGI assigns natural-language concepts to visual tokens and performs concept-level selection before they enter the language model. This makes the retained and discarded information directly interpretable, enables application- or user-controlled pruning through semantic concepts, and avoids model-specific pruning layers, retention-ratio schedules, and repeated modifications of the language backbone.

3 METHODOLOGY

Notations: We consider an autoregressive vision-language model M comprising a visual encoder M_v, a projection module M_{pr}, and a large language model (LLM) decoder M_{LLM} of dimension d with L layers. Given an input image I, the visual encoder divides it into N patches and encodes them into visual features. These features are then projected by M_{pr} into an embedding space of dimension d, yielding the visual representation $f_v = M_{pr}(M_v(I))$, where $f_v \in \mathbb{R}^{N\times d}$. Similarly, a textual prompt is tokenized into T tokens and embedded into a textual representation $f_t \in \mathbb{R}^{T\times d}$. The visual and textual representations are then concatenated along the sequence dimension to form the input to the language decoder, i.e., $f = [f_v; f_t]$, where $f \in \mathbb{R}^{(N+T)\times d}$. The language decoder M_{LLM} autoregressively processes f and produces a hidden representation h_t^l for each layer l at each generation step t. The final layer hidden representation h_t^L at each step t is mapped to the vocabulary distribution of size V through an unembedding linear layer $W_U \in \mathbb{R}^{d\times V}$. Note that during both the prefill and generation phases, due to the autoregressive nature of LLMs, image tokens attend to neither prompt nor generated tokens, and are therefore not affected by prompt variation.

A high-level overview of our method is presented in Figure 2. Our first goal is to annotate each visual token with a natural language text, enabling us to interpret which visual tokens to remove or retain, and to gain simple and greater control over the generation process. We define a concept as a semantically meaningful word (*e.g.,* ear, coat, waves) and seek to associate each visual token with such a concept. To achieve this, we exploit the model's shared vision–language representation space and apply the logit lens technique (Nostalgebraist, 2020). Originally introduced for LLMs, the logit lens has also been widely used to interpret vision–language models (Neo et al., 2025) and mitigate their hallucinations (Jiang et al., 2025). In VLMs, the Logit Lens technique leverages the hidden layer visual token representations, which are well aligned with the language embedding space. By projecting these representations through the unembedding layer W_U, we obtain a probability distribution over the LLM's vocabulary space. We then select the vocabulary token with the highest

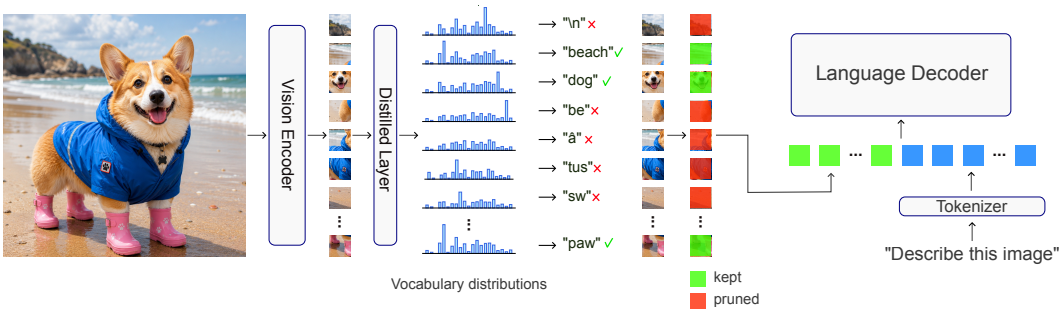


Figure 2: Overview of the proposed CORGI approach. Given an image processed by the vision encoder, visual tokens are selectively passed to the language decoder by retaining only those whose hidden states project onto semantically meaningful concepts in the vocabulary space, optionally via a lightweight MLP termed the distiller layer, trained to predict the vocabulary distribution.

probability to obtain a textual interpretation of each visual token. Notably, this requires an additional forward pass through all visual tokens in M_{LLM} to extract their concept annotations before determining which tokens should be pruned, which would add additional complexity. In order to address this problem, we train a simple MLP $MLP_{dis} \in \mathbb{R}^d \rightarrow \mathbb{R}^V$ to distill the last layer visual tokens vocabulary distribution, directly to f_v using the Kullback–Leibler (KL) Divergence loss:

$$\mathcal{L}_{\text{KL}} = \sum_{i=1}^{N} \text{KL}\left(\text{softmax}\left(W_U^T\, h_i^{L,v}\right) \,\middle\|\, \text{softmax}\left(MLP_{\text{dis}}\ (f_{v,i})\right)\right), \tag{1}$$

where $h^{L,v}$ are the visual tokens representations at the last layer of M_{LLM}. Since MLP_{dis} is trained from the VLM's own distribution, this setting becomes fully self-supervised and does not require labels. We select the final layer L as the concepts extracted from other layers $< L$ are not semantically meaningful and give lower performance (Appendix A.2.2). Also note that training MLP_{dis} is optional and is used both to reduce computational cost (FLOPs) and to improve concept annotation quality. Indeed, distillation offers an inherent denoising mechanism (Zhao, 2023; Kaiser et al., 2022), as confirmed by our comparison with and without distillation (Table 5 in the Appendix), where omitting it leads to a slight performance drop. In the case where we do not use the distillation layer, we select visual tokens at the prefill stage once, and all subsequent generation steps rely on those selected tokens (Appendix A.2.1).

Following Hinton et al. (2015), we distill soft targets. The predicted vocabulary probabilities capture informative relationships among the tokens in the vocabulary that a visual token may represent. We also provide ablation studies on training with hard labels (Appendix A.2.1), and show that soft ones provide a smoothing effect that mitigates overfitting to noisy hard ones (Yuan et al., 2020; Furlanello et al., 2018). Once trained, MLP_{dis} can directly infer the concept associated with each visual token from f_v, without requiring a forward pass through M_{LLM}. The concept associations, denoted as C, can therefore be obtained before the visual tokens are passed to M_{LLM} by running f_v through MLP_{dis} and taking the token with the highest probability. That is, $C = \arg\max\left(MLP_{\text{dis}}(f_v)\right)$ and $C \in \mathbb{R}^N$. We then convert the predicted token back into ordinary, readable text using the model's own native tokenizer conversion routine. Further implementation details are provided in Appendix A.3.

Pruning Visual Tokens: The procedure described above assigns a semantic concept annotation to each of the N visual tokens in f_v, making every token directly interpretable. As a result, visual token pruning can be formulated as a concept-level selection problem: tokens corresponding to meaningful concepts are retained, while the remaining tokens are discarded. This provides a simple yet flexible framework that allows pruning strategies to be defined according to semantic concepts. One can also define concepts relevant to a specific application. For example, in autonomous driving, one may want to retain concepts corresponding to the driving scene such as "road", "pedestrian", and "sign" rather than the background scene such as "hills", "sky" and "clouds". We validate this controllability aspect in Section 4.3. As a general-purpose pruning strategy, we introduce a simple approach termed *Most Common*. Specifically, we maintain a vocabulary of 20K of the most frequently occurring English words and retain only visual tokens whose predicted concept belongs to

this vocabulary, while pruning all others. Common function words (e.g., *the*, *a*, and *an*) are excluded from the vocabulary, as they carry no semantic information. The underlying intuition is that visual tokens corresponding to semantically meaningful concepts are typically decoded into common lexical items, whereas less informative or noisy tokens tend to be mapped by the unembedding layer to punctuation symbols (e.g., ".". ","), formatting tokens (e.g., "\n"), unknown or malformed tokens (e.g., "uż"), or other non-semantic vocabulary entries. The proposed strategy removes nearly 80% of the visual tokens while incurring only a slight degradation in performance.

Redundancy-aware Pruning: In order to have more control over pruning and reach a higher pruning ratio, we further eliminate redundant visual tokens. Specifically, tokens with the *same* predicted concept are selected, and only the top-k tokens with the highest prediction confidence (i.e., the highest logits produced by MLP_{dis}) are retained, thereby removing redundant tokens. This strategy preserves the most representative visual evidence for each concept while removing duplicate semantic information. This additional redundancy-aware pruning enables pruning ratios of up to 95% with only a marginal degradation in generation performance. We emphasize that this redundancy-aware pruning step is optional and is applied only when a more aggressive pruning ratio is desired. Setting $k = \infty$ disables this additional pruning stage, such that all visual tokens associated with the same predicted concept are retained. The above process yields Z kept tokens, where $Z << N$. The Z kept tokens are then fed to the language decoder M_{LLM}, with no layer-wise dropping schedules as in previous approaches (Wu et al., 2026; Arif et al., 2025; Chen et al., 2024; Xing et al., 2024; Singh et al., 2026). The impact of the choice of k on generation performance is analyzed in Section 4.4. The results show consistent performance across all benchmarks and metrics, suggesting that k can be defined globally and generally, thereby alleviating the need for additional tuning.

4 Experiments

Evaluation tasks: We conduct experiments across a diverse set of tasks and domains to demonstrate the effectiveness of our approach. We place particular emphasis on image captioning, as it reflects a more real world scenario (Kim et al., 2026) as it generates several token compare to VQA, which generate a single token. For image captioning, we evaluate on DetailCaps (Dong et al., 2024), COCO (Chen et al., 2015) and LLaVA-Bench (Liu et al., 2023). We also evaluate on image-based visual question answering using the GQA benchmark, to enable direct comparison with existing token compression methods that primarily report results on VQA tasks. We further extend our evaluation to the video domain, assessing VQA performance on TGIF-QA (Jang et al., 2017), MSVD-QA, and MSRVTT-QA (Xu et al., 2017). Finally, to validate our approach beyond natural images and video, we evaluate on medical report generation using MIMIC-CXR (Johnson et al., 2019), CheXpert Plus (Chambon et al., 2024), and IU X-Ray (Demner-Fushman et al., 2016).

For captioning and report generation tasks, we report METEOR (Banerjee & Lavie, 2005), ROUGE-L (Lin, 2004), and BERTScore (Zhang et al., 2020) as common metrics; image captioning is additionally evaluated with CLIPScore, RefCLIPScore (Hessel et al., 2021) and CLAIR (Chan et al., 2023), an LLM as a judge captioning metric (see Appendix A.3.1), while medical report generation includes RadGraph-F1 (Jain et al., 2021) to capture clinical accuracy. For VQA, we report accuracy.

Models: All experiments are conducted using Qwen2.5-VL-7B (Bai et al., 2025) as our primary backbone, which offers a good trade-off between model size and performance among open-source VLMs. To assess the scalability of our approach, we additionally evaluate on image captioning using the larger Qwen2.5-VL-32B variant. Finally, to enable comparison with earlier token compression methods, we also experiment with LLaVA-1.5-7B (Liu et al., 2024) on image VQA.

Baselines: As CORGI does not require compression-aware re-training or fine-tuning of the LVLM, we compare it against baselines with a similar setup. We select Duet-VLM (Singh et al., 2026), the current state of the art method, as well as the base model without any token compression applied. For the image VQA experiments, we additionally compare against FastV (Chen et al., 2024), HiRED (Arif et al., 2025), PruMerge (Shang et al., 2025), SparseVLM (Zhang et al., 2024), PyramidDrop (Xing et al., 2024), and VisionZip (Yang et al., 2025b), whose results are taken from Duet-VLM work (Singh et al., 2026), where all methods were evaluated on a LLaVA backbone.

In all comparisons, methods are evaluated at equivalent pruning ratios. For our method, this ratio is controlled by the parameter k; for Duet-VLM, it is determined by the dominant and contextual token

Table 1: Comparison of CORGI against Duet-VLM on captioning datasets

		Metrics						Efficiency		
Dataset	Method	METEOR	ROUGE-L	BERT-S	CLIP-S	RefCLIP-S	CLAIR	GFLOPs ↓	Pruned(%) ↑	Tokens Gen.
Qwen2.5-VL-7B										
DetailCaps	Baseline	21.0	27.7	84.1	32.2	32.0	75.53	2616	0.0	173.3
	Duet-VLM [CVPR 2026]	14.3	22.1	80.6	29.3	30.4	46.60	245	84.5	209.4
	CORGI (Ours)	**18.5**	**25.7**	**83.1**	**30.3**	**31.0**	**58.23**	395	85.2	156.4
COCO	Baseline	21.1	35.5	82.1	32.5	32.7	70.66	2642	0.0	24.1
	Duet-VLM [CVPR 2026]	18.0	34.6	81.2	31.1	31.9	**63.97**	534	82.8	24.6
	CORGI (Ours)	**19.6**	**35.3**	**81.8**	**31.3**	**32.0**	63.08	543	83.1	19.9
LLaVA-Bench	Baseline	23.3	28.4	82.6	32.6	32.0	82.55	7560	0.0	240.8
	Duet-VLM [CVPR 2026]	13.0	24.3	79.0	**31.1**	**31.2**	**61.41**	1265	80.0	277.5
	CORGI (Ours)	**20.9**	**24.7**	**81.0**	31.0	31.1	60.83	1487	80.9	234.6
Qwen2.5-VL-32B										
DetailCaps	Baseline	21.8	20.3	78.2	31.7	31.8	78.24	12370	0.0	499.7
	Duet-VLM [CVPR 2026]	18.4	18.6	76.6	27.8	29.6	35.28	805	92.9	428.5
	CORGI (Ours)	**19.6**	**19.0**	**77.0**	**28.9**	**30.2**	**52.96**	1097	93.0	460.2
COCO	Baseline	19.4	33.6	81.0	31.6	32.2	68.95	12517	0.0	22.0
	Duet-VLM [CVPR 2026]	15.7	29.8	79.5	30.0	31.3	56.16	1509	91.8	18.4
	CORGI (Ours)	**17.2**	**31.7**	**80.2**	**30.2**	**31.4**	**57.05**	1506	92.1	19.9
LLaVA-Bench	Baseline	21.0	18.1	77.7	31.7	31.5	82.97	35697	0.0	540.3
	Duet-VLM [CVPR 2026]	**19.5**	**16.7**	**77.0**	**31.1**	**31.2**	**65.55**	2337	94.0	515.4
	CORGI (Ours)	18.8	15.5	75.8	30.5	30.9	54.00	2513	94.5	507.3

parameters, which set the number of visual tokens passed to the language model. The exact parameter values used for each method and task combination are provided in Table 7 of the Appendix. Finally, in all results tables, the theoretical GFLOPs and pruning ratio are reported alongside each result; their computation is detailed in Sections A.3.2 and A.3.3 of the Appendix.

4.1 Results

Image Captioning: Table 1 reports image captioning results with Qwen2.5-VL-7B and Qwen2.5-VL-32B. Compared to the baseline, both pruning methods show a moderate performance drop, expected under aggressive token compression. At equivalent pruning ratios, CORGI consistently outperforms Duet-VLM on DetailCaps and COCO across nearly all metrics and both model sizes. On LLaVA-Bench, results are more mixed: with the 7B model, our method leads on METEOR, ROUGE-L, and BERTScore while Duet-VLM performs better on CLAIR, CLIPScore, and RefCLIPScore; with the 32B model, Duet-VLM outperforms our method across all metrics. Overall, the 32B results confirm that CORGI scales to larger models while maintaining its advantage on the majority of captioning benchmarks.

Beyond raw performance, CORGI offers a qualitatively different operating point: token selection is fully interpretable through concept annotations and directly controllable, as unwanted concepts can be explicitly excluded from the accepted concept set, preventing associated visual patches from entering the model's input, as we demonstrate in Section 4.3. The only pruning hyperparameter is k, which can be set to ∞ to obtain a strong baseline with no redundancy filtering. In contrast, Duet-VLM requires model-specific layer schedules and multiple clustering hyperparameters.

Interestingly, absolute metric values tend to be lower for the 32B model than for the 7B across all methods, including the baseline. We attribute this to differences in output length: on DetailCaps and LLaVA-Bench, the 32B model generates responses approximately twice as long as the 7B model, whereas on COCO, both models produce comparably short captions, as the dataset instructions explicitly prompt for brief descriptions. Reference-based metrics such as METEOR, ROUGE-L, and BERTScore are sensitive to this discrepancy: a longer generation may accurately describe image content not covered by the ground truth, yet still incur a penalty, as these metrics measure overlap with a fixed reference rather than factual correctness. Longer outputs thus tend to dilute metric scores independently of generation quality, which complicates cross-model comparison on datasets where output length varies substantially.

Table 2: GQA accuracy comparison of our approach against Duet-VLM on Qwen2.5-VL-7B and LLaVA-1.5-7B.

		Metrics	Efficiency	
Model	Method	Acc (%)	GFLOPs ↓	Pruned (%) ↑
Qwen2.5-VL-7B	Baseline	60.8	2737	0.0
	Duet-VLM	52.1	468	89.3
	CORGI (Ours)	**52.9**	465	89.9
LLaVA-1.5-7B	Baseline	61.9	4010	0.0
	FastV	46.1	–	88.9
	HiRED	54.6	644	88.9
	PruMerge	51.9	644	88.9
	SparseVLM	52.7	–	88.9
	PyramidDrop	47.5	–	88.9
	VisionZip	55.1	644	88.9
	Duet-VLM	**56.7**	646	88.9
	CORGI (Ours)	53.6	644	88.9

Table 3: Comparison of CORGI against Duet-VLM on video QA datasets with Qwen2.5-VL-7B.

		Metrics	Efficiency	
Dataset	Method	Acc (%) ↑	GFLOPs ↓	Pruned (%) ↑
TGIF-QA	Baseline	57.5	3310	0.0
	Duet-VLM	48.9	489	92.1
	CORGI (Ours)	**49.2**	530	92.3
MSVD-QA	Baseline	41.0	12741	0.0
	Duet-VLM	28.0	656	93.5
	CORGI (Ours)	**31.9**	737	94.0
MSRVTT-QA	Baseline	19.0	3050	0.0
	Duet-VLM	14.9	460	91.4
	CORGI (Ours)	**16.3**	438	91.7

Visual Question Answering: Table 2 reports the accuracy on GQA for both Qwen2.5-VL-7B and LLaVA-1.5-7B. On Qwen2.5-VL-7B, our approach outperforms Duet-VLM, demonstrating the effectiveness of our method. On LLaVA-1.5-7B, our method remains competitive with most existing token compression approaches. A key distinction of our approach compared to existing methods is that it leverages the alignment between visual tokens and the LLM's embedding space, rather than relying solely on vision encoder representations or attention-based pruning. As a result, the relative performance of our method is influenced by the quality of this cross-modal alignment in the underlying model. We discuss this further in Section 4.2, where we show that this alignment is substantially weaker in LLaVA than in Qwen, which explains why our method does not surpass state-of-the-art performance on the former.

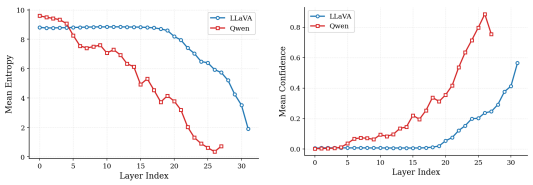
Video Question Answering: Table 3 reports results on video question answering. CORGI consistently outperforms Duet-VLM across all three datasets, demonstrating that our approach generalizes well to video inputs, where token compression is particularly critical given the large number of visual tokens generated by multi-frame sequences.

Table 4: Comparison of our approach against Duet-VLM on medical report generation datasets with Qwen2.5-VL-7B.

		Metrics					Efficiency	
Dataset	Method	BLEU-1	METEOR	ROUGE-L	BERTScore	RadGraph	GFLOPs ↓	Pruned (%) ↑
MIMIC-CXR	Baseline	5.3	12.1	9.7	80.2	9.0	9204	0.0
	Duet-VLM	5.8	11.6	10.2	80.5	8.5	733	93.4
	CORGI (Ours)	**6.0**	**12.3**	**10.2**	**80.8**	**9.0**	678	94.0
CheXpert Plus	Baseline	2.0	10.6	8.3	77.9	2.1	9200	0.0
	Duet-VLM	2.3	10.7	9.0	78.3	2.1	733	93.4
	CORGI (Ours)	**2.4**	**12.2**	**9.7**	**78.5**	**2.8**	677	94.0
IU X-Ray	Baseline	2.7	11.1	7.2	80.5	11.1	2604	0.0
	Duet-VLM	2.3	**12.3**	7.2	80.4	**12.5**	584	88.5
	CORGI (Ours)	**2.7**	11.3	**7.3**	**81.0**	11.5	562	89.7

Medical Report Generation: Table 4 reports results on medical report generation. Our approach consistently outperforms Duet-VLM on MIMIC-CXR and CheXpert Plus across all metrics, including RadGraph F1, which captures clinical accuracy. Notably, on both datasets, both methods also surpass the baseline on several metrics. We hypothesize that medical images, being high-resolution by nature, produce a large number of visual tokens, a significant portion of which may correspond to uninformative background regions; token compression thus acts as a denoising step, discarding redundant patches and encouraging the model to focus on clinically relevant image regions. On IU X-Ray, results are more mixed, with no clear winner across metrics.

4.2 Image-Text Alignment Analysis



(a) (b)

Figure 3: Layer-wise alignment between the visual and textual space of the VLM, measured on the visual patches selected by logit lens, for LLaVA-1.5-7B and Qwen2.5-VL-7B.

Our approach relies on the quality of the logit lens, which is the alignment between visual patch tokens and the model's vocabulary embedding space, to identify informative visual tokens. To better understand how the quality of this alignment influences the relative performance of our approach compared to DuetVLM, we analyze it layer by layer across 300 random samples from Visual Genome (Krishna et al., 2017). For each visual patch selected by our token filtering at each layer, we project its hidden state into the vocabulary space and compute two measures: 1) the entropy of the resulting distribution, which reflects how uncertain the model is about which token the patch corresponds to, with lower entropy indicating a more peaked, semantically precise distribution, and 2) the confidence, defined as the probability assigned to the most likely token in that distribution. Note that the confidence is also the normalized vector similarity between the visual token embedding and the textual token embedding of the most likely token (i.e., a column of W_U).

Figure 3 reveals a striking difference between the two models. In LLaVA, both entropy and confidence remain nearly constant up to layer 19, suggesting that visual patch tokens carry little semantic meaning in the vocabulary embedding space throughout the early and middle layers. Only beyond layer 19 does the entropy begin to decrease and confidence increase, indicating that alignment emerges late in the network, contrary to Qwen which aligment emerges at layer 4. From layer 4 onwards, Qwen consistently exhibits lower entropy and higher confidence than LLaVA, demonstrating a stronger overall alignment between visual patch tokens and the vocabulary embedding space. This is consistent with recent findings showing that stronger language models tend to exhibit higher visual alignment (Ruthardt et al., 2024). Since our method selects tokens based on this alignment signal, it naturally benefits from models where alignment is well-developed across layers, which explains why our approach stands out more clearly from the state of the art on Qwen than on LLaVA. As vision-language models continue to improve, stronger backbone alignment will further benefit our approach, making it well-suited for deployment on current and future state-of-the-art models.

4.3 Concept-level Intervention

A key advantage of our approach is that, since token selection is interpretable, it is possible to intervene to control which visual concepts the model attends to during generation. Specifically, by removing one or more intended concepts, the visual patches associated with these concepts are excluded from the kept set, preventing the model from generating content related to them.

Figure 4 illustrates this on two examples. In all cases, the intervened output no longer mentions the suppressed concept(s), while remaining fluent, coherent, and consistent with the output of the baseline. Notably, this holds even when suppressing multiple concepts simultaneously, as shown in the second example where both *wine* and *bottles* are removed. Furthermore, the intervention proves effective even in challenging cases where the visual evidence is spatially diffuse or subtle: in the first example, the bridge appears small and partially occluded in the background, yet its associated patches are successfully excluded, leading the model to describe only the foreground scene. This controllability stands in contrast to existing token compression methods, which operate as black boxes and do not offer any mechanism for human intervention in the token selection process. Additional examples are shown in Figure 7 of the Appendix.

4.4 Token Redundancy

The parameter k determines the number of visual tokens retained for each semantic concept when multiple visual tokens are associated with the same concept, allowing us to adjust the pruning ratio to match those of the methods we compare against. Figure 5 shows the impact of k on captioning performance across DetailCaps, COCO, and LLaVA-Bench. Performance improves sharply for most of the metrics between $k = 1$ and $k = 3$, then stabilizes for larger values, confirming that tokens associated with the same concept are largely redundant and that a small number of representatives



Baseline output: A motorcyclist rides along a dirt path, surrounded by lush greenery and a misty mountain backdrop, with a bridge in the distance.

Initial pruning output: A motorcyclist rides along a dirt path in a lush, green rural area, with a bridge visible in the background.

Intervention → removing "bridge" tokens

rural ~~bridge~~ man vegetation riding path helmet dirt

Intervened output: A lone motorcyclist rides along a dirt path surrounded by lush greenery, navigating through a rural landscape.



Baseline output: Vintage motorcycle displayed in a wine cellar, blending classic style with rustic charm.

Initial pruning output: Vintage motorcycle displayed in a wine cellar, blending classic style with rustic charm.

Intervention → removing "wine" and "bottles" tokens

bike display tiles cellar ~~wine~~ engine vintage ~~bottles~~

Intervened output: Vintage motorcycle displayed in a rustic setting, showcasing its classic design and timeless appeal.

Figure 4: Qualitative examples of CORGI's interpretability through concept intervention. For each example, we show the baseline output (no pruning), the initial pruned output, and an intervened output obtained by manually removing one or more concepts from the extracted concept set, demonstrating the causal influence of individual concepts on the generated caption.

per concept is sufficient to capture the relevant visual information. Overall, these results validate the use of k as a reliable control parameter for reaching a target pruning ratio without significantly affecting performance. Across all benchmarks and metrics, we find that $k = 3$ offers the best trade-off between pruning percentage and accuracy retention, suggesting a robust, general-purpose choice of k across different settings.

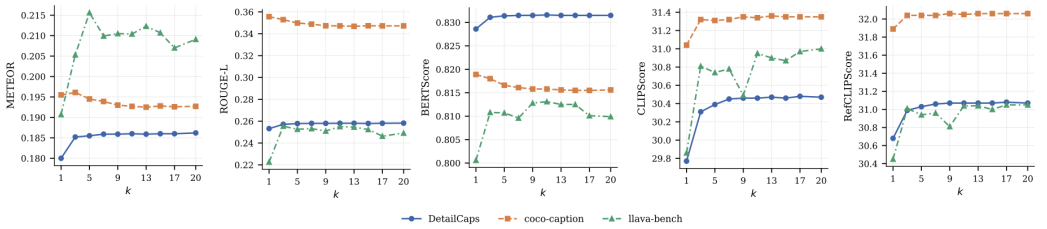


Figure 5: Ablation study on k across three captioning datasets. Each subplot shows the evolution of one metric as k increases from 1 to 20.

5 Conclusion

We presented CORGI, an interpretable and controllable visual token pruning framework for Vision-Language Models. By projecting visual token representations through the language model's unembedding layer, CORGI assigns natural language concept annotations to visual tokens prior to language model processing, enabling semantically grounded token selection without layer-wise pruning schedules or complex hyperparameter tuning. Experiments across image captioning, visual question answering, video understanding, and medical report generation demonstrate that CORGI often outperforms the current state of the art at equivalent pruning ratios. Beyond efficiency gains, token selection is fully transparent and unwanted visual concepts can be explicitly suppressed to control model generation, making CORGI a step toward more trustworthy and auditable multimodal systems, with natural extensions to domain-specific applications such as clinical imaging or autonomous driving. Finally, as with any research work, our study has certain limitations, which are clearly acknowledged in Section A.1 of the Appendix.

AI DISCLOSURE

We have not used generative AI tools for generating synthetic datasets, proposing or refining hypotheses, designing research methodology or experiments, implementing methods, translation, cleaning and reformatting datasets, qualitative and thematic data analysis, or interpreting results. Developing theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proofs, and writing proofs are not applicable to this work.

Additionally, we used generative AI tools to create the dog image used in Figures 1 and 2, to help formulate and refine the wording of certain parts of the paper, and to assist in adapting DUET code to new datasets. All AI-assisted work was reviewed by the authors: AI-generated text was checked for accuracy and revised accordingly, AI-assisted code was verified and tested for correctness, and the AI-generated image was reviewed for consistency with the paper's content. We take responsibility for the final content of this work.

REFERENCES

Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4190–4197, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.385.

Kazi Hasan Ibn Arif, JinYi Yoon, Dimitrios S Nikolopoulos, Hans Vandierendonck, Deepu John, and Bo Ji. Hired: Attention-guided token dropping for efficient inference of high-resolution vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 1773–1781, 2025.

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

Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pp. 65–72, 2005.

Pierre J. Chambon, Jean-Benoit Delbrouck, Thomas Sounack, Shih-Cheng Huang, Zhihong Chen, Maya Varma, Steven Quoc Hung Truong, Chu The Chuong, and Curtis P. Langlotz. Chexpert plus: Augmenting a large chest x-ray dataset with text radiology reports, patient demographics and additional image formats. *ArXiv*, abs/2405.19538, 2024.

David M Chan, Suzanne Petryk, Joseph E Gonzalez, Trevor Darrell, and John Canny. Clair: Evaluating image captions with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13638–13646, 2023.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, 2024. URL https://api.semanticscholar.org/CorpusID:268358224.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

Dina Demner-Fushman, Marc D Kohli, Marc B Rosenman, Sonya E Shooshan, Laritza Rodriguez, Sameer Antani, George R Thoma, and Clement J McDonald. Preparing a collection of radiology examinations for distribution and retrieval. *Journal of the American Medical Informatics Association*, 23(2):304–310, 2016.

Hongyuan Dong, Jiawen Li, Bohong Wu, Jiacong Wang, Yuan Zhang, and Haoyuan Guo. Benchmarking and improving detail image caption. *arXiv preprint arXiv:2405.19092*, 2024.

Tommaso Furlanello, Zachary Chase Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *International Conference on Machine Learning*, 2018.

Zhiyu Guo, Hidetaka Kamigaito, and Taro Watanabe. Attention score is not all you need for token importance indicator in KV cache reduction: Value also matters. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 21158–21166, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.1178.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 7514–7528, 2021.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531, 2015.

Saahil Jain, Ashwin Agrawal, Adriel Saporta, Steven Truong, Du Nguyen Duong, Tan Bui, Pierre Chambon, Yuhao Zhang, Matthew P. Lungren, Andrew Y. Ng, Curtis Langlotz, and Pranav Rajpurkar. Radgraph: Extracting clinical entities and relations from radiology reports. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.

Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2758–2766, 2017.

Nicholas Jiang, Anish Kachinthaya, Suzanne Petryk, and Yossi Gandelsman. Interpreting and editing vision-language representations to mitigate hallucinations. In *The Thirteenth International Conference on Learning Representations*, 2025.

Alistair EW Johnson, Tom J Pollard, Seth J Berkowitz, Nathaniel R Greenbaum, Matthew P Lungren, Chih-ying Deng, Roger G Mark, and Steven Horng. Mimic-cxr, a de-identified publicly available database of chest radiographs with free-text reports. *Scientific data*, 6(1):317, 2019.

Timo Kaiser, Lukas Ehmann, Christoph Reinders, and Bodo Rosenhahn. Blind knowledge distillation for robust image classification. *ArXiv*, abs/2211.11355, 2022.

Jiwan Kim, Kibum Kim, Wonjoong Kim, Byung-Kwan Lee, and Chanyoung Park. Why and when visual token pruning fails? a study on relevant visual information shift in mllms decoding. *arXiv preprint arXiv:2604.12358*, 2026.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. Attention is not only a weight: Analyzing transformers with vector norms. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7057–7075, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.574.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pp. 74–81, 2004.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 26296–26306, 2024.

Clement Neo, Luke Ong, Philip Torr, Mor Geva, David Krueger, and Fazl Barez. Towards interpreting visual information processing in vision-language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Nostalgebraist. Interpreting GPT: The logit lens. https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens, August 2020. LessWrong blog post, accessed July 27, 2026.

Jona Ruthardt, Gertjan J Burghouts, Serge Belongie, and Yuki M Asano. Better language models exhibit higher visual alignment. *arXiv preprint arXiv:2410.07173*, 2024.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22857–22867, 2025.

Aditya Kumar Singh, Hitesh Kandala, Pratik Prabhanjan Brahma, Zicheng Liu, and Emad Barsoum. Duet-vlm: Dual stage unified efficient token reduction for vlm training and inference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17651–17660, 2026.

Hao Wu, Yingqi Fan, Dai Jinyang, Junlong Tong, Yunpu Ma, and Xiaoyu Shen. Hidrop: Hierarchical vision token reduction in MLLMs via late injection, concave pyramid pruning, and early exit. In *The Fourteenth International Conference on Learning Representations*, 2026.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, et al. Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. *arXiv preprint arXiv:2410.17247*, 2024.

Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. Video question answering via gradually refined attention over appearance and motion. In *Proceedings of the 25th ACM international conference on Multimedia*, pp. 1645–1653, 2017.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. Visionzip: Longer is better but not necessary in vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19792–19802, 2025b.

Li Yuan, Francis E. H. Tay, Guilin Li, Tao Wang, and Jiashi Feng. Revisiting knowledge distillation via label smoothing regularization. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3902–3910, 2020.

Tianyi Zhang, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SkeHuCVFDr.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, et al. Sparsevlm: Visual token sparsification for efficient vision-language model inference. *arXiv preprint arXiv:2410.04417*, 2024.

Jiachen Zhao. Student as an inherent denoiser of noisy teacher. *ArXiv*, abs/2312.10185, 2023.

A APPENDIX

A.1 LIMITATIONS

Dependence on the Common Word Vocabulary: A key design choice of CORGI is the use of a fixed vocabulary of 20K common English words[1] to filter semantically meaningful visual tokens. This vocabulary was selected for its broad coverage and immediate availability, and provides a strong general-purpose baseline across diverse tasks. In our experiments, we applied this same vocabulary uniformly across all datasets, including medical ones; notably, even a general-purpose word list such as the 20K vocabulary already contains medically relevant terms (*lung*, *heart*, *disease*, *tissue*, *abnormal*, *tumor*) which are sufficient to guide meaningful pruning on chest X-ray datasets. Moreover, for a generalist VLM, a highly specialized vocabulary may actually be counterproductive: if the model has not been trained on clinical terminology, its extracted concepts are unlikely to match domain-specific terms, resulting in few or no matches and degraded pruning. A general vocabulary thus strikes a better balance for non-specialized models, anchoring pruning on concepts the model itself is likely to produce. Substituting this general-purpose vocabulary with a domain-specific concept list could further improve performance in specialized settings when using a domain-adapted model. Importantly, the pruning framework is designed to accommodate arbitrary concept vocabularies: the 20K list can be straightforwardly extended with domain-specific terminology, or replaced entirely by a task-specific concept list curated for a target application. For non-English settings, an analogous substitution with a language-appropriate vocabulary would similarly address the English-language assumption. We therefore view the current vocabulary choice as a practical default rather than a fundamental constraint, with domain-specific adaptation being a natural next step for deployment in clinical or other specialized contexts.

MLP training: Our method reduces computational overhead by training a distilled MLP, MLP_{dis}, to approximate the probability distribution produced by the final VLM layer. While our pruning method does not depend on this distilled MLP and can operate without it, MLP_{dis} is still helpful to achieve better computational savings. A limitation of this approach is that the distillation process must be performed separately for each VLM to which we wish to apply pruning.

A.2 ABLATION STUDY

A.2.1 DISTILLATION

As described in Section 3, we train a lightweight linear layer MLP_{dis} to predict concept annotations directly from the projected visual features f_v, avoiding a forward pass through M_{LLM} at inference time. We compare two training objectives for MLP_{dis}: our proposed soft distillation via KL divergence, which supervises MLP_{dis} to reproduce the full vocabulary distribution produced by W_U, versus hard label training via cross-entropy, where the supervision is collapsed to the single most probable concept, discarding inter-concept probability mass. We additionally evaluate CORGI *w/o Distillation*, where concept annotations are obtained directly via W_U after a full forward pass through M_{LLM}, without training any additional module. This setting demonstrates that our approach can alleviate training MLP_{dis} at the expense of increased computation, demonstrating that our the MLP is optional rather than a core dependency.

Results in Table 5 show that soft distillation yields consistent and substantial improvements across all datasets and text-quality metrics. Hard label training, by contrast, performs on par with the no-distillation setting across all benchmarks, confirming that collapsing the supervision to a single argmax fails to capture the semantic ambiguity inherent to visual patches, which soft targets preserve through the full vocabulary distribution. The gains brought by MLP_{dis} are more pronounced on lexical overlap metrics such as METEOR and ROUGE-L, while semantic alignment metrics remain largely unaffected, consistent with the denoising effect of soft targets discussed in Section 3.

The *w/o Distillation* setting incurs a moderate performance drop compared to the distilled variant, yet demonstrates that CORGI remains effective without training the distiller layer. However, training MLP_{dis} is computationally inexpensive, requiring only a small amount of data and a lightweight optimization, while yielding substantial inference-time savings; we therefore recommend its use in practice.

[1] https://github.com/first20hours/google-10000-english/blob/master/20k.txt

Table 5: Comparison of distilled variants on Qwen2.5-VL-7B. For these experiments, k is set to ∞

Dataset	Method	METEOR	ROUGE-L	BERT-S	CLIP-S	RefCLIP-S	GFLOPs ↓
DetailCaps	w/o Distillation	15.8	23.3	81.3	30.6	31.1	3238
	Distilled Linear Hard	15.9	23.3	81.4	30.6	31.1	630
	Distilled Linear	18.6	25.8	83.2	30.5	31.1	590
COCO-Caption	w/o Distillation	15.8	31.8	79.7	30.5	31.6	3468
	Distilled Linear Hard	15.7	31.7	79.7	30.4	31.5	761
	Distilled Linear	19.3	34.7	81.5	31.4	32.1	737
LLaVA-Bench-in-the-Wild	w/o Distillation	11.9	20.3	77.5	30.8	31.0	9585
	Distilled Linear Hard	12.3	22.6	78.0	30.3	30.7	1575
	Distilled Linear	20.9	24.7	81.0	31.0	31.1	1487

A.2.2 Layer Selection

The logit lens readout can in principle be applied at any layer of the LLM. To determine which layer yields the most semantically meaningful concepts, we measure the overlap between the concepts extracted at each layer and the ground truth object instances and attributes from 300 randomly sampled images of the Visual Genome dataset (Krishna et al., 2017). As shown in Figure 6, the overlap is zero at all layers except the last, where it reaches its maximum. We therefore use the last layer for all our experiments.

To further validate this choice, we note that the entropy and confidence analysis of Figure 3 shows slightly better alignment metrics at the penultimate layer than at the last one for Qwen. This raises the question of whether concepts extracted at that layer might still be relevant, even if they do not match Visual Genome annotations. To investigate this, we compare captioning performance on Qwen2.5-VL-7B using the last layer (layer 28) versus the penultimate layer (layer 27), without token redundancy filtering ($k = \infty$). As shown in Table 6, performance drops drastically across all datasets and metrics when using the penultimate layer, confirming that the last layer produces the most task-relevant concepts and justifying our design choice.

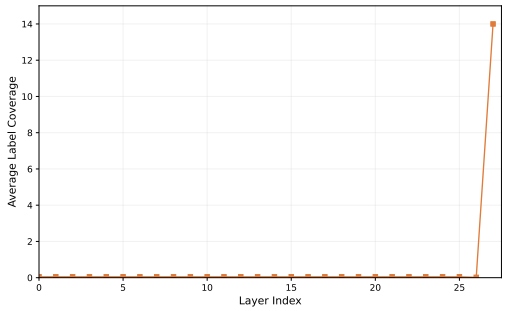


Figure 6: Average number of Visual Genome object instances and attributes covered by the concepts extracted at each layer of Qwen2.5-VL-7B.

Table 6: Comparison of our approach (penultimate layer vs. last layer) on three captioning datasets with Qwen2.5-VL-7B-Instruct, without token filtering ($k = \infty$).

Dataset	Layer	METEOR	ROUGE-L	BERT-S	CLIP-S	RefCLIP-S
DetailCaps	penultimate	6.6	14.9	72.8	17.5	22.4
	last	**18.6**	**25.8**	**83.2**	**30.5**	**31.1**
COCO	penultimate	5.1	17.9	71.8	18.2	23.4
	last	**19.3**	**34.7**	**81.5**	**31.4**	**32.1**
LLaVA-Bench	penultimate	9.2	14.4	72.2	22.2	25.8
	last	**20.9**	**24.7**	**81.0**	**31.0**	**31.1**

A.3 Implementation Details

BPE tokenization: Although BPE tokenization can represent a word using multiple pieces, frequent words and meaningful lexical stems are commonly also represented as standalone vocabulary entries. CORGI therefore does not require reconstruction of the complete surface word for a visual token to express a useful semantic concept. For instance, a concept related to eating may project to a lexical token such as eat, which independently belongs to a common word, whereas a fragment such as "ing" would normally be rejected. Thus, CORGI seeks an interpretable lexical concept rather than requiring recovery of the complete inflected surface form.

Linear Layer Training: To avoid the computational overhead of running a full forward pass through the LLM at inference time, we train a lightweight student network to approximate the logit

lens output directly from the visual projector features. The student network consists of two linear layers with a LayerNorm and a GELU activation in between. The first layer projects d dimensions into $d \times 2$, while the second projects the $d \times 2$ dimensions into V dimensions, matching the model vocabulary distribution size. The MLP is trained to match the vocabulary distribution produced by the last LLM layer at visual token positions via KL divergence loss with temperature $T = 1.0$. A separate student network is trained for each backbone: Qwen2.5-VL-7B, Qwen2.5-VL-32B, and LLaVA-1.5-7B. All models are trained on 80k images from the COCO 2014 training dataset (Chen et al., 2015), split into 90% training and 10% validation, using the Adam optimizer with an initial learning rate of 10^{-4} and a cosine annealing schedule down to 10^{-6} over 5 epochs. The best checkpoint is selected based on validation loss. For the layer ablation study (Section 4.4), a separate checkpoint is trained for Qwen2.5-VL-7B using the penultimate layer as the teacher signal instead of the last layer. For the hard target ablation, we train using cross-entropy with teacher-generated hard pseudo-labels, as in Equation 2.

$$\mathcal{L}_{\text{dis}} = -\sum_{i=1}^{N} \log \left[\text{softmax}\left(MLP_{\text{dis}} f_{v,i}\right)\right]_{\hat{y}_i}, \qquad \hat{y}_i = \arg\max_c \left[W_U h_i^{L,v}\right]_c. \tag{2}$$

Inference Setup All models are loaded in `bfloat16` with standard attention implementation and run on a single NVIDIA H200 GPU (140 GB). For Qwen2.5-VL models, the resolution and patch size are determined dynamically based on the input image size, up to a maximum resolution fixed at $1280 \times 28 \times 28$ pixels, which in practice never exceeds 1280 visual tokens across our evaluation datasets. For LLaVA-1.5-7B, the input resolution is fixed at 336×336 pixels with a patch size of 14×14 pixels, yielding a fixed number of 576 visual tokens. For video datasets, each clip is reduced to 8 frames sampled uniformly along its temporal axis. GIF clips (TGIF-QA) are fully decoded before uniform sampling; MP4 clips (MSVD-QA, MSRVTT-QA) are read with `decord` and 8 frame indices are drawn uniformly with `np.linspace`.

Generation is performed with greedy decoding and a maximum of 1024 new tokens. All models are used in their instruction-tuned variants: `Qwen/Qwen2.5-VL-7B-Instruct`, `Qwen/Qwen2.5-VL-32B-Instruct`, and `llava-hf/llava-1.5-7b-hf`.

For Duet-VLM, we use the layer-wise pruning schedules provided in their official codebase[2]: for Qwen2.5-VL-7B, tokens are pruned at layers 14 and 21 with ratios $[1.0, 0.5, 0.25]$; for Qwen2.5-VL-32B, at layers 32 and 48 with the same ratios; and for LLaVA-1.5-7B, at layers 16 and 24 with ratios $[1.0, 0.5, 0.0]$.

A.3.1 CLAIR EVALUATION

For the captioning datasets (Table 1), we report CLAIR scores following the evaluation formulation and prompt design of Chan et al. (2023). CLAIR is an LLM-as-a-judge captioning metric that scores a candidate caption against one or more reference captions without requiring access to the image. For COCO, we provide the complete set of reference captions; for DetailCaps and LLaVA-Bench, we use the available ground-truth reference. All CLAIR scores are computed using Qwen3-32B as the judge model.

Our scoring prompt follows the original CLAIR formulation:

> You are trying to tell if a candidate set of captions is describing the same image as a reference set of captions. Candidate set: [candidate]. Reference set: [references]. On a precise scale from 0 to 100, how likely is it that the candidate set is describing the same image as the reference set?

The judge is instructed to return a numeric score in the interval $[0, 100]$.

A.3.2 GFLOPS COMPUTATION

We report theoretical prefill GFLOPs to compare computational efficiency independently of generation length. For a transformer layer processing a sequence of S tokens, with hidden dimension d

[2] `https://github.com/AMD-AGI/Duet-VLM`

Table 7: Summary of hyperparameters and pruning ratios per experiment. For Duet-VLM, *dom.* and *ctx.* are hyperparameters controlling the number of visual patches fed to the model after the vision encoder. For CORGI (Ours), k is the maximum number of patches associated with the same concept that are retained; $k = \infty$ means all patches that pass the vocabulary-match filter are kept.

Model	Task	Dataset	Duet-VLM (dom., ctx.)	CORGI (k)	Pruned (%) DUET / CORGI
Qwen2.5-VL-7B	Captioning	DetailCaps	(24, 4)	$k = 3$	84.5 / 85.2
		COCO	(71, 13)	$k = 3$	82.8 / 83.1
		LLaVA-Bench	(191, 34)	$k = \infty$	80.0 / 80.9
	Image VQA	GQA	(44, 8)	$k = 1$	89.3 / 89.9
	Video VQA	TGIF-QA	(31, 5)	$k = 1$	92.1 / 92.3
		MSVD-QA	(79, 14)	$k = 1$	93.5 / 94.0
		MSRVTT-QA	(44, 8)	$k = 1$	91.4 / 91.7
	Report Generation	MIMIC-CXR	(101, 19)	$k = 3$	93.4 / 94.0
		CheXpert Plus	(101, 19)	$k = 3$	93.4 / 94.0
		IU X-Ray	(44, 8)	$k = 7$	88.5 / 89.7
Qwen2.5-VL-32B	Captioning	DetailCaps	(11, 2)	$k = 1$	92.9 / 93.0
		COCO	(34, 6)	$k = 1$	91.8 / 92.1
		LLaVA-Bench	(58, 10)	$k = 3$	94.0 / 94.5
LLaVA-1.5-7B	Image VQA	GQA	(72, 30)	$k = 3$	88.9 / 88.9

and intermediate *Feed Forward Network* size d_{ff}, the FLOPs are estimated as:

$$\mathcal{F}(S) = 4Sd^2 + S^2d + 3Sd \cdot d_{ff} \tag{3}$$

For CORGI and the baseline, all L layers process the same sequence of Z kept visual tokens and T text tokens, giving a total prefill cost of $L \cdot \mathcal{F}(Z + T)$ FLOPs.

For Duet-VLM, which progressively reduces the number of visual tokens across stages inside the LLM, the prefill cost is computed per stage: each group of K_i consecutive layers sees $\lfloor N_0 \cdot r_i \rfloor$ visual tokens according to the pruning schedule $\{r_i\}$, giving $\sum_i K_i \cdot \mathcal{F}(\lfloor N_0 \cdot r_i \rfloor + T)$ FLOPs, where $N_0 = N_{\text{dom}} + N_{\text{ctx}}$ is the number of visual tokens entering the LLM, with N_{dom} and N_{ctx} being the dominant and contextual token counts as defined in Duet-VLM (Singh et al., 2026).

The text token count T is computed per sample by tokenizing the instruction with the model's tokenizer. All values are reported in GFLOPs ($\times 10^9$ FLOPs).

GFLOPs are recomputed using this procedure for all experiments involving CORGI and Duet-VLM. Note that these GFLOPs account exclusively for the LLM prefill stage, excluding any pre-LLM processing such as the vision encoder and upstream token reduction steps.

For the additional baselines reported in Table 2 (HiRED, PruMerge, VisionZip), GFLOPs are computed using the same formula as CORGI, since these methods perform token selection entirely before the LLM, meaning all L layers process the same reduced sequence of Z visual tokens and T text tokens. For methods that prune tokens inside the LLM (FastV, SparseVLM, PyramidDrop), GFLOPs are not reported: since we did not re-implement these approaches, the exact hyperparameters used to obtain the target pruning ratio are unknown, and the per-layer token schedule required for an accurate FLOPs estimate cannot be determined.

A.3.3 Pruning Ratio Computation

For Qwen2.5-VL models, the number of visual tokens varies across samples depending on the input image resolution. We therefore compute the pruning ratio as:

$$\text{Pruned}(\%) = \frac{1}{|\mathcal{D}|} \sum_{i \in \mathcal{D}} \frac{N_i - Z_i}{N_i} \times 100 \tag{4}$$

where $\mathcal{D}$ is the evaluation dataset, N_i is the total number of visual tokens and Z_i the number of kept tokens for sample i. For Duet-VLM, since visual tokens are progressively dropped across LLM stages, Z_i is computed as the average number of visual tokens seen per layer, weighted by the number of layers in each stage. Since our approach achieves more aggressive minimum pruning ratios than those covered by the configurations originally proposed in Duet-VLM (Singh et al.,

2026), we identify additional configurations via binary search over $N_0 = N_{\text{dom}} + N_{\text{ctx}}$, targeting the desired pruning percentage on each dataset, while maintaining the same dominant-to-contextual ratio as the original configurations. The full set of experimental parameters, including dominant and contextual token counts for Duet-VLM and k values for our approach, along with their associated pruning ratios per dataset and model, are reported in Table 7.

A.4 QUALITATIVE EXAMPLES

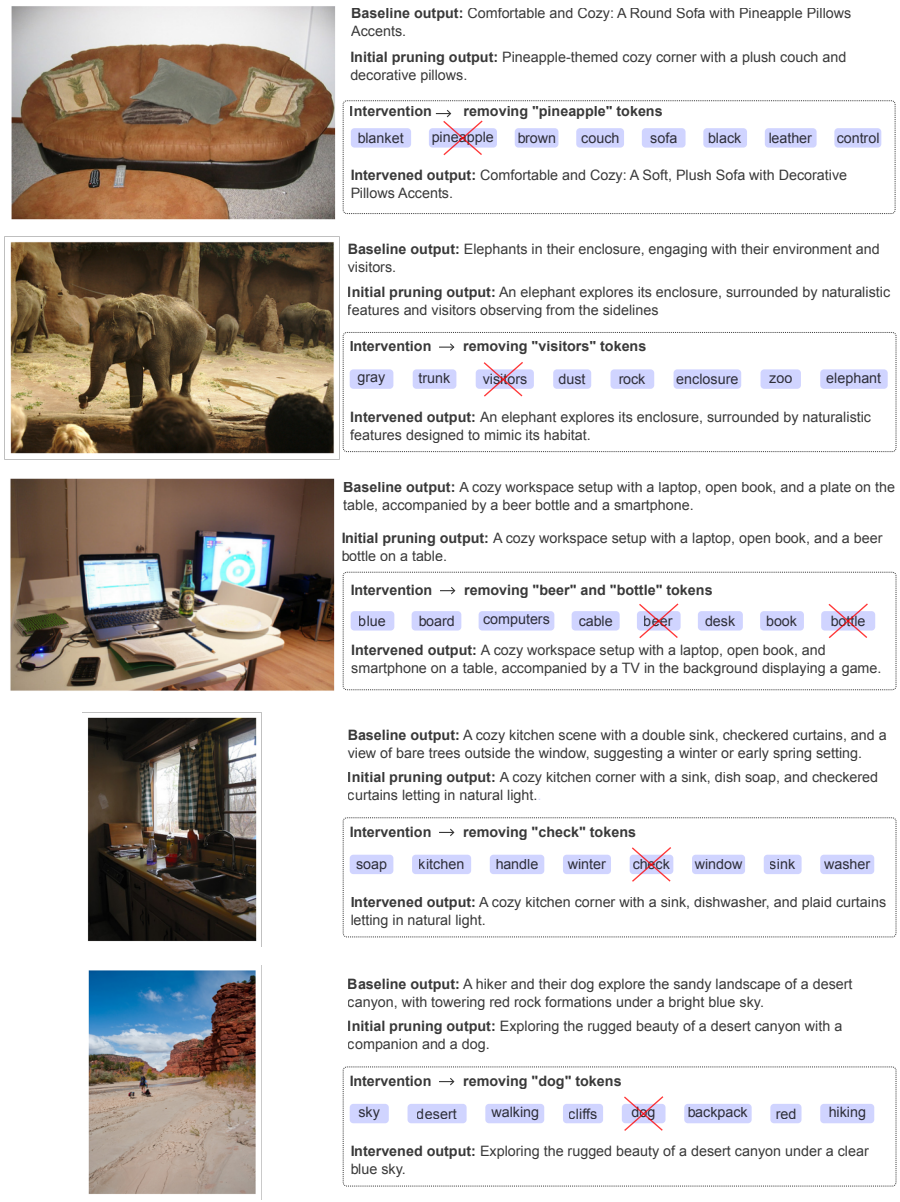


Figure 7: Qualitative examples of CORGI's interpretability through concept intervention. For each example, we show the baseline output (no pruning), the initial pruned output, and an intervened output obtained by manually removing one or more concepts from the extracted concept set, demonstrating the causal influence of individual concepts on the generated caption.