Enabling Visual Reasoning in Text-Only LLMs via Symbolic Scene Descriptions

Wenzhuo Lei, Mufan Cao, Chaorong Ye, Yi Xu, Cheng Chen*
Tongji University, Shanghai, China
Emails: 2452002@tongji.edu.cn, 235278@tongji.edu.cn, yechaorong@tongji.edu.cn
2351441@tongji.edu.cn, 2351445@tongji.edu.cn
*Corresponding author

***Abstract*—We study how far a text-only Large Language Model (LLM) can go on visual question answering when vision is reduced to a compact symbolic interface. Our framework, DeepYOLO, uses an off-the-shelf detector to convert an image into a grammar-constrained scene report of object labels and coarse locations, which is then consumed by a text-only LLM without multimodal pre-training. On a stratified 5k subset of VQAv2, DeepYOLO reaches 42.3% VQA accuracy, outperforming a scene-graph serialization baseline (38.2%) while using 2.5× fewer prompt tokens. Replacing detections with ground-truth objects raises performance to 44.5%, indicating that detector quality, rather than symbolic reasoning, is the main bottleneck. The resulting pipeline is modular, auditable, and attractive for privacy-sensitive or resource-constrained deployment.**

***Index Terms*—Large Language Models, Object Detection, YOLO, DeepSeek, Multi-modal AI**

I. Introduction

Large vision-language models usually inject dense visual tokens directly into the language model [1]–[4]. This design is effective, but it is expensive, hard to audit, and makes perception errors difficult to separate from reasoning errors. We instead ask a simpler question: *can a text-only LLM answer visual questions if the image is first reduced to a compact symbolic scene description for real-image VQA benchmarks such as VQAv2 [5], [6]?*

This question matters beyond efficiency alone. A discrete interface between perception and reasoning offers a different systems trade-off from end-to-end multimodal alignment: it localizes failures, reduces the amount of information sent to the language model, and exposes an intermediate representation that can be inspected by humans. In privacy-sensitive or edge settings, these properties are often as important as raw benchmark accuracy, especially when recent multimodal models continue to rely on hundreds of visual tokens or large visual encoders [7], [8].

Our answer is **DeepYOLO**, a loose-coupled neuro-symbolic pipeline in which YOLO extracts object labels and coarse positions, and DeepSeek-style reasoning models [9], [10] operate over the resulting grammar-constrained scene report. Because the interface is discrete and textual, it keeps the evidence trail explicit, avoids multimodal training, and allows the detector and LLM to be upgraded independently. Rather than proposing a new foundation model, we focus on the quality of the *interface* between vision and language: what information must be preserved, how compact that interface can be, and where performance is actually lost.

Contributions. First, we show that a text-only LLM can reach 42.3% VQA accuracy on a stratified 5k subset of VQAv2 using only symbolic scene reports. Second, under the same detector+text constraint, we show that a compact 3×3 grid serialization outperforms an explicit scene graph baseline (38.2%) while using 2.5× fewer prompt tokens, suggesting that concise representations can be better reasoning interfaces than more verbose symbolic structures. Third, by replacing detector outputs with ground-truth objects, we obtain 44.5% accuracy and thereby isolate the dominant bottleneck: perception quality rather than downstream symbolic reasoning. Together, these results position DeepYOLO as a practical and auditable alternative for resource-limited visual reasoning.

II. Related Work

Continuous-alignment VLMs. Most modern VLMs, including CLIP-style pretraining, Flamingo, BLIP-2, LLaVA, and GPT-4V, project visual features into the language model and achieve strong VQA performance [1]–[4], [11]. Their strength comes from dense visual grounding learned through multimodal pre-training or alignment. However, this same design raises token cost, blurs the boundary between perception and reasoning, and makes it difficult to ask whether a compact symbolic interface might already be sufficient for a useful subset of visual reasoning tasks.

Neuro-symbolic VQA. Prior neuro-symbolic systems separate perception and reasoning with explicit symbolic states or executable programs [12], [13]. Related tool-oriented work also shows that compositional visual reasoning can be achieved by orchestrating detectors, segmenters, and text modules without end-to-end multimodal training [14]. These systems established the value of structured intermediate representations, but they often relied on specialized program executors, synthetic datasets, or narrowly scoped symbolic vocabularies. DeepYOLO keeps the separation principle but replaces the hand-designed executor with a general-purpose LLM, allowing the same symbolic interface to be tested on natural-image VQA without retraining a dedicated reasoning module.

Efficient and edge-oriented models. Another line of work reduces multimodal cost by shrinking the visual encoder or

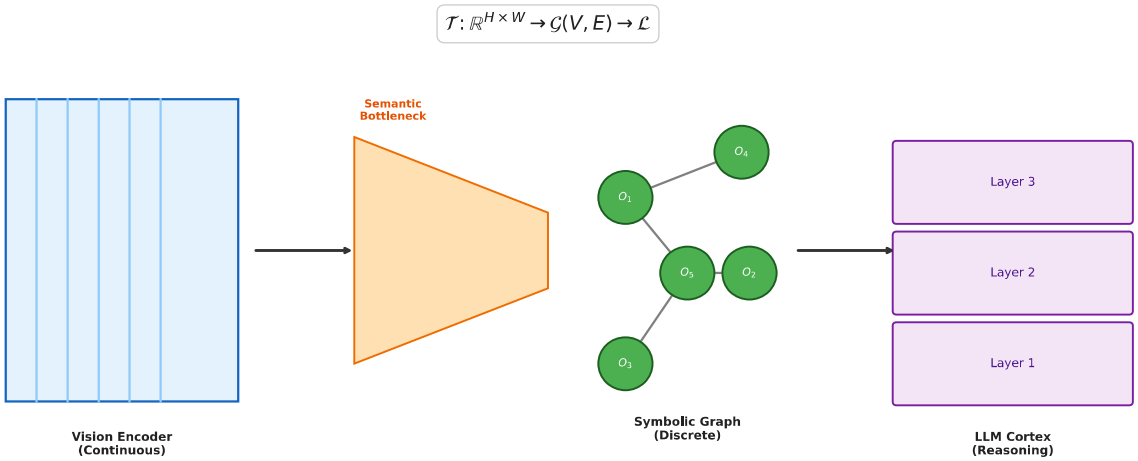

Fig. 1. DeepYOLO converts detector outputs into a compact scene report that a text-only LLM can reason over.

by building tool-augmented reasoning pipelines [15], [16]. DeepYOLO is complementary to this direction. Instead of compressing the continuous visual stack, it changes the interface itself: the detector runs locally, the LLM receives a short textual report, and token usage stays predictable. The perception side also sits naturally within the long line of explicit detectors, from Faster R-CNN to DETR and newer YOLO variants [17]–[20]. Our closest baseline is therefore not a full VLM, but an alternative detector+text interface such as verbose scene-graph serialization.

III. Theoretical Framework

We interpret DeepYOLO through the Information Bottleneck (IB) principle [21]. Let X denote the image, S the symbolic scene report, and Y the answer. An ideal interface keeps $I(S;Y)$ high while making $I(X;S)$ small: it preserves the information needed for the task while discarding nuisance variation. In DeepYOLO, this compression is implemented explicitly by mapping pixels to a small set of object labels and coarse spatial bins.

From a rate-distortion perspective, the scene report acts as a semantic quantizer. It removes textures, lighting, and other high-dimensional details, but retains object-centric cues that a language model can manipulate reliably. This view explains both the strengths and the limits of the approach: as long as the answer depends mostly on object identity and coarse layout, the compressed interface can remain sufficient; once the task depends on color, OCR text, or fine-grained geometry, the bottleneck becomes lossy in a task-relevant way. The resulting accuracy–efficiency frontier is therefore determined largely by what the detector and serializer preserve, not only by the reasoning capacity of the LLM.

IV. Methodology

DeepYOLO is a two-stage pipeline that lets a text-only LLM answer visual questions from an object-centric symbolic representation. It combines an explicit detector front-end in the YOLO family [19], [20] with a reasoning-oriented language model [10], [22].

A. Scene Report Construction

Given an image I, YOLOv8n produces detections $D = \{(c_i, s_i, b_i)\}_{i=1}^{N}$ containing a category label, confidence, and bounding box. We discard low-confidence detections, map each box center to one of nine cells in a 3×3 grid, and serialize the remaining objects into short phrases such as "person at Center" or "car at Bottom-Left." This deterministic grammar keeps the interface compact and stable across images while retaining the kind of explicit localization signal that detector-based systems expose naturally [17], [18].

In practice, the symbolic interface is intentionally minimal: it keeps only category and coarse location, and omits raw pixels, dense patch tokens, and most detector metadata. This makes the representation easy to inspect and keeps prompt length predictable even when the upstream detector changes.

B. LLM Reasoning

The scene report is concatenated with the user question and passed to DeepSeek-R1 [10] using deterministic decoding ($T = 0$). The prompt instructs the model to answer only from the provided symbols and to return `unknown` when the required evidence is absent. Because the representation is textual, the perception and reasoning modules remain fully decoupled.

This separation also lets us define controlled variants without changing the downstream reasoner. For the SceneGraph baseline, we keep the same detector and LLM but replace the compact report with a more verbose object-and-relation

Algorithm 1 DeepYOLO Inference

Require: Image I, question Q, threshold τ

$D \leftarrow \text{YOLO}(I)$
$S \leftarrow \emptyset$
for all $(c_i, s_i, b_i) \in D$ **do**
  if $s_i > \tau$ **then**
    $g_i \leftarrow \text{GridLoc}(b_i)$
    add "c_i at g_i" to S
  end if
end for
$P \leftarrow \text{Format}(S, Q)$
$R \leftarrow \text{LLM}(P)$
return R

serialization. For the Oracle setting, we replace detector outputs with ground-truth object annotations while preserving the same prompting and decoding procedure. These matched variants make it possible to attribute performance differences to the symbolic interface itself rather than to changes in the backbone model.

C. Why It Is Efficient

Unlike a standard VLM, DeepYOLO performs visual processing outside the language model and usually sends fewer than one hundred prompt tokens to the LLM. The language-model cost therefore scales with the symbolic report length rather than hundreds of image tokens, while the intermediate representation stays human-auditable.

Formally, if a conventional VLM exposes N_{img} visual tokens to the language model, its attention cost scales with $\mathcal{O}((N_{\text{img}} + N_{\text{text}})^2)$. DeepYOLO instead pushes vision into the detector and sends only N_{sym} symbolic tokens, where $N_{\text{sym}} \ll N_{\text{img}}$. The dominant language-model cost becomes $\mathcal{O}((N_{\text{sym}}+N_{\text{text}})^2)$, which explains why compact scene reports reduce both prompt budget and reasoning latency.

In addition, DeepYOLO moves bandwidth away from raw images and toward a narrow textual interface. This shift is useful in practical deployments: the detector can run on-device, only the scene report needs to cross the model boundary, and each answer is traceable back to explicit symbols rather than hidden visual embeddings.

V. Experiments

A. Experimental Setup

We evaluate on a stratified 5k subset of the VQAv2 validation split [5], [6] with answer-type proportions of 40% yes/no, 30% number, and 30% other. The detector is YOLOv8n and the reasoning model is DeepSeek-R1 with deterministic decoding. To isolate the effect of the symbolic interface, we compare three settings under the same detector+text pipeline: **DeepYOLO**, an explicit **SceneGraph** serialization of the same detections, and an **Oracle** setting that replaces detections with COCO ground-truth objects. We report keyword accuracy, VQA soft accuracy, BLEU-4, average LLM latency, and average prompt tokens.

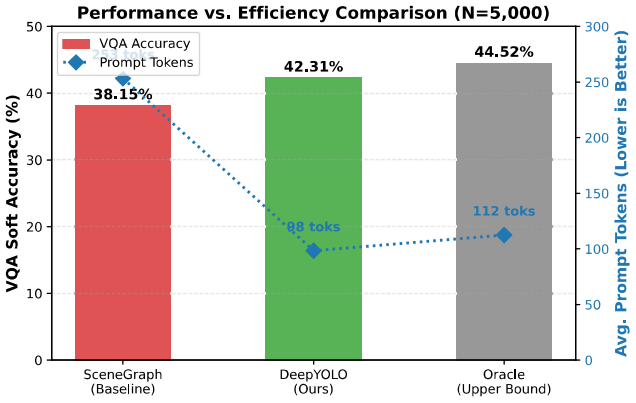

Fig. 2. DeepYOLO improves over SceneGraph while using far fewer prompt tokens, and remains close to the Oracle upper bound.

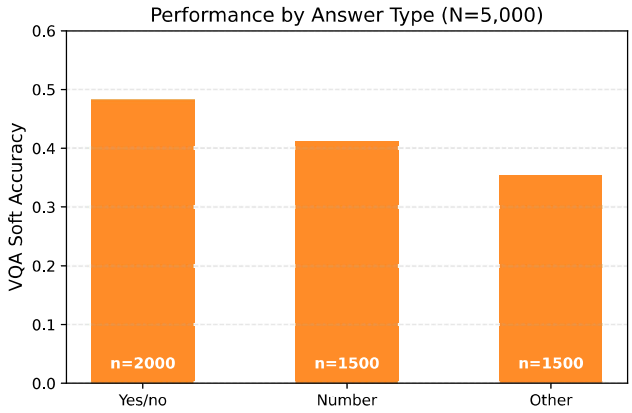

Fig. 3. Accuracy by VQAv2 answer type on the 5k evaluation split.

B. Main Results

Table I shows that DeepYOLO reaches 42.3% VQA accuracy, outperforming the more verbose SceneGraph baseline by 4.1 points while using 2.5× fewer prompt tokens. The Oracle setting improves to 44.5%, leaving only a 2.2-point gap and indicating that detector quality, rather than symbolic reasoning, is the main bottleneck.

C. Additional Analysis

The symbolic bottleneck is intentionally lossy, so attribute-heavy questions remain difficult when color or text is absent from the report. Lightweight plugins partially mitigate this issue, as summarized in Table II. On a separate 500-example pilot used for diagnostics, we observe three dominant error modes: missed detections, missing attributes, and coarse spatial resolution in crowded scenes.

These results clarify the intended comparison. DeepYOLO is not meant to beat large end-to-end VLMs that benefit from dense visual features and multimodal pre-training. Instead, it tests whether a compact symbolic interface is sufficient for useful visual reasoning in a text-only LLM. Under that criterion, the strongest reference is the Oracle setting: the

TABLE I
MAIN EVALUATION ON VQAV2 VALIDATION SUBSET (ANSWER-TYPE STRATIFIED, $N = 5{,}000$)

Method	Keyword Acc	VQA Acc	BLEU-4	LLM Lat. (s, avg)	Prompt Tok. (avg)
DeepYOLO (compact)	**39.2%**	**42.3%**	28.5	1.05	**98.4**
SceneGraph baseline	35.4%	38.2%	22.1	1.15	253.2
Oracle GT objects (max 50)	41.8%	44.5%	30.2	1.10	112.5

DeepYOLO outperforms the SceneGraph baseline in both accuracy and efficiency, and closely approaches the Oracle GT upper bound, establishing the symbolic interface's sufficiency for visual reasoning.

TABLE II
LIGHTWEIGHT ATTRIBUTE PLUGINS ON ATTRIBUTE-HEAVY SUBSETS.

Subset	N	Base	Plugin	Δ
Color	375	0.1%	9.5%	+9.4
Text/OCR	136	2.2%	2.9%	+0.7

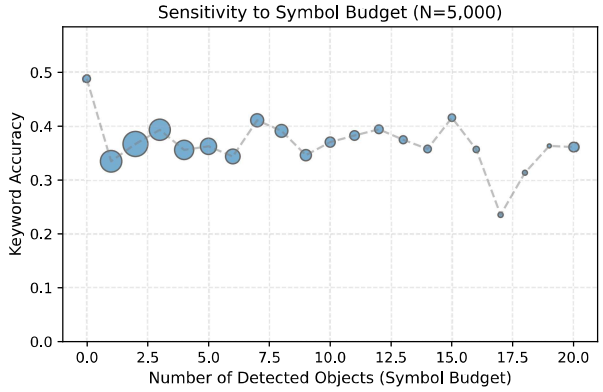

Fig. 4. Sensitivity to symbol budget. Moderate numbers of symbols provide the best trade-off between coverage and noise.

narrow 2.2-point gap shows that most remaining headroom lies in perception rather than in the reasoning module.

The 5k evaluation is also large enough to support stable comparison. With $N = 5{,}000$, the Wilson 95% confidence interval around the main accuracy estimate is approximately $\pm 1.37\%$, so the 4.1-point advantage over SceneGraph is unlikely to be explained by subset-level sampling noise alone.

Figure 4 suggests that DeepYOLO works best with a moderate symbol budget. If too few detections are retained, the scene report under-specifies the image; if too many are serialized, irrelevant objects begin to distract the LLM. This observation further motivates the compact grammar used in DeepYOLO rather than a maximally verbose serialization of detector outputs.

D. Efficiency

DeepYOLO is also communication-efficient. On the pilot subset, the average JPEG image size is about 154 KB, whereas the text sent to the LLM averages about 469 B, a 337× reduction in transmitted content. Median YOLO latency is 59.8 ms per image, and average LLM latency remains near one second, with typical prompts containing fewer than one hundred tokens.

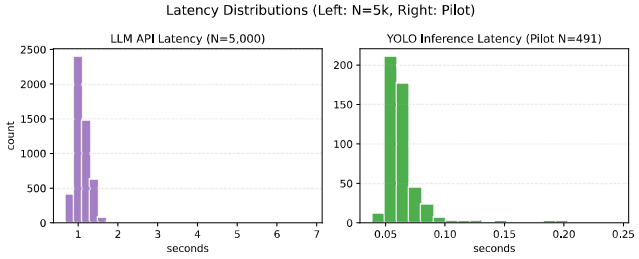

Fig. 5. Latency distributions for the LLM stage and the YOLO perception stage.

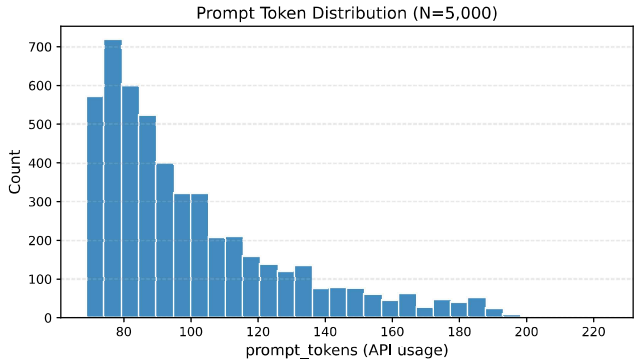

Fig. 6. Prompt token distribution on the 5k evaluation set. Most DeepYOLO prompts remain compact despite scene variability.

Figure 5 shows that the detector contributes a small and stable fraction of end-to-end delay, whereas the LLM dominates runtime variance. This is consistent with the design goal of shifting visual computation into a lightweight local front-end while keeping the language model input compact.

Token usage remains similarly well behaved. Although the number of detections varies across images, most prompts stay tightly concentrated around the low-hundreds range because the grammar suppresses unnecessary metadata and relation enumeration. This is the practical reason DeepYOLO can be both interpretable and cheap to query.

E. Implementation Details

For reproducibility, the detector is YOLOv8n with confidence threshold $\tau = 0.25$ and standard non-maximum suppression. Input images are resized to 640×640, and each surviving box is mapped to a 3×3 grid using the box center. The resulting scene report keeps only object category and

TABLE III
REPRESENTATIVE CASES FROM THE PILOT ANALYSIS.

Question	Top Symbols	Prediction	GT	Diagnosis
Is there a clock?	clock×1	Yes	yes	Correct existence reasoning
What activity are these men doing?	person×2, skateboard×1	skating	skateboarding	Correct action grounding
Is there a boat in the water?	bench×1	No	yes	Detector missed the boat
How many people are standing in a row on the right?	person×14, surfboard×4	0	4	Grid too coarse for fine spatial layout

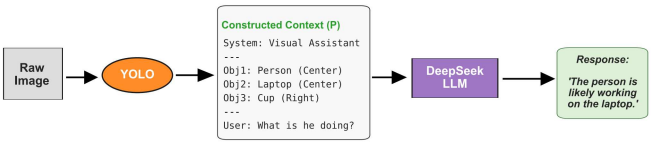

Fig. 7. The intermediate scene report provides an explicit audit trail for success and failure cases.

coarse location, which intentionally trades away fine-grained attributes for a shorter and more stable prompt.

The reasoning model is queried with deterministic decoding. We use a short-answer prompt that instructs the model to answer only from detected symbols and to output `unknown` when the required evidence is missing. Quantitative evaluation uses temperature 0, maximum output length 32 tokens, and direct scoring against the VQAv2 references.

The 5k main split is drawn from the VQAv2 validation set with answer-type stratification (40% yes/no, 30% number, 30% other) and a fixed random seed of 42. The 500-example pilot subset (actual N=491 after filtering out images with detection failures, corrupted inputs, or missing logs) is used only for profiling and qualitative diagnosis. We log per-example detections, prompts, responses, and scores so that both successful and failed cases remain auditable.

VI. DISCUSSION

Why the compact interface works. The main lesson of our study is that a short symbolic prompt can be a better interface for a text-only LLM than a more verbose scene graph built from the same detections. We attribute this to reduced relational noise: the grid report gives the model object presence and coarse geometry, but avoids flooding it with pairwise relations that increase token cost and distract reasoning. The Oracle result strengthens this interpretation, because once the objects are correct, DeepYOLO already recovers 95% of Oracle performance.

What the Oracle gap implies. The remaining gap to the Oracle setting is small but informative. It suggests that, for this task configuration, the reasoning model is not the dominant limitation; instead, errors arise primarily when the symbolic front-end omits or corrupts the evidence that the LLM needs. This is exactly the kind of separation that end-to-end VLMs make difficult to observe. In other words, DeepYOLO is useful not only as an efficient pipeline, but also as a measurement tool for studying where visual reasoning systems fail.

Practical implications. A discrete detector-to-language interface is attractive in deployment regimes where privacy, debuggability, or system modularity matter. The detector can be upgraded independently of the LLM, scene reports can be logged and audited, and raw images do not need to be sent to a remote multimodal API. These properties make the approach especially relevant for edge or compliance-sensitive applications where an interpretable intermediate state is a systems requirement rather than a cosmetic benefit, complementing recent work on compact multimodal and on-device models [15], [16], [23].

Limitations and outlook. At the same time, the symbolic bottleneck defines a clear ceiling. Object-only symbols do not encode color, text, or fine spatial structure, so performance drops on attribute-heavy and crowded scenes. This limitation is also a design opportunity: instead of reverting immediately to dense visual embeddings, the system can be extended with lightweight, auditable modules such as OCR, attribute tags, depth cues, or adaptive spatial discretization [24], [25]. A promising next step is therefore not to abandon symbolic interfaces, but to enrich them just enough to recover missing evidence while preserving compactness and traceability. More broadly, this connects to the classic symbol-grounding question: the challenge is not whether symbols are useful, but how much grounded visual detail they must carry for a target task [26].

VII. CONCLUSION

DeepYOLO shows that a text-only LLM can perform non-trivial visual reasoning when the image is converted into a compact symbolic scene report. On VQAv2-5k, the method reaches 42.3% VQA accuracy, outperforms a more verbose scene-graph baseline, and stays close to the 44.5% Oracle upper bound. These results suggest that, in this setting, the dominant challenge is perception quality rather than symbolic reasoning itself.

More broadly, our findings argue that multimodal capability does not always require dense multimodal alignment inside the LLM. A carefully designed symbolic interface can already deliver useful reasoning, cleaner failure attribution, and a substantially smaller token budget. The resulting detector-to-text pipeline is modular, auditable, and well suited to privacy-aware visual reasoning systems.

AI USE STATEMENT

Generative AI was used only for language polishing, including grammar correction and wording refinement of author-written text. It was not used for method design, experiments,

result analysis, or scientific claims; all content was reviewed and approved by the authors.

APPENDIX A
ADDITIONAL IMPLEMENTATION DETAILS

We use YOLOv8n pretrained on COCO with confidence threshold 0.25, NMS IoU 0.45, and input resolution 640×640. The LLM is queried through an OpenAI-compatible API with temperature 0.0, top-p 0.9, and maximum output length 32 tokens. All experiments use zero-shot prompting without fine-tuning. The VQAv2-5k subset is sampled with fixed seed 42 using a 40%/30%/30% stratification over answer types. Prompt token counts are taken from the API-reported `prompt_tokens` and include both the system prompt and constructed user message.

Each surviving detection is converted into a short phrase by mapping its box center to a 3×3 semantic grid. The exported scene report keeps only object category and coarse location, and intentionally omits raw coordinates, confidence values, and pairwise relations in order to keep prompts short and stable across images.

The short-answer prompt instructs the model to answer only from the provided symbols and to output `unknown` when the required evidence is absent. For auditability, we log per-example detections, prompts, model outputs, and scores; the 500-example pilot split (actual N=491 after filtering out images with detection failures, corrupted inputs, or missing logs) is used only for profiling and qualitative diagnosis, while the 5k split is used for quantitative comparison.

REFERENCES

[1] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PmLR, 2021, pp. 8748–8763.

[2] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds *et al.*, "Flamingo: a visual language model for few-shot learning," *Advances in neural information processing systems*, vol. 35, pp. 23 716–23 736, 2022.

[3] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," in *International conference on machine learning*. PMLR, 2023, pp. 19 730–19 742.

[4] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," *Advances in neural information processing systems*, vol. 36, pp. 34 892–34 916, 2023.

[5] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "Vqa: Visual question answering," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2425–2433.

[6] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh, "Making the v in vqa matter: Elevating the role of image understanding in visual question answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6904–6913.

[7] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[8] G. Team, R. Anil, S. Borgeaud, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, K. Millican *et al.*, "Gemini: a family of highly capable multimodal models," *arXiv preprint arXiv:2312.11805*, 2023.

[9] X. Bi, D. Chen, G. Chen, S. Chen, D. Dai, C. Deng, H. Ding, K. Dong, Q. Du, Z. Fu *et al.*, "Deepseek llm: Scaling open-source language models with longtermism," *arXiv preprint arXiv:2401.02954*, 2024.

[10] D. Guo, D. Yang, H. Zhang, J. Song, P. Wang, Q. Zhu, R. Xu, R. Zhang, S. Ma, X. Bi *et al.*, "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning," *arXiv preprint arXiv:2501.12948*, 2025.

[11] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[12] K. Yi, J. Wu, C. Gan, A. Torralba, P. Kohli, and J. Tenenbaum, "Neural-symbolic vqa: Disentangling reasoning from vision and language understanding," *Advances in neural information processing systems*, vol. 31, 2018.

[13] J. Mao, C. Gan, P. Kohli, J. B. Tenenbaum, and J. Wu, "The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision," *arXiv preprint arXiv:1904.12584*, 2019.

[14] T. Gupta and A. Kembhavi, "Visual programming: Compositional visual reasoning without training," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 14 953–14 962.

[15] X. Chu, L. Qiao, X. Lin, S. Xu, Y. Yang, Y. Hu, F. Wei, X. Zhang, B. Zhang, X. Wei *et al.*, "Mobilevlm: A fast, strong and open vision language assistant for mobile devices," *arXiv preprint arXiv:2312.16886*, 2023.

[16] Z. Liu, C. Zhao, F. Iandola, C. Lai, Y. Tian, I. Fedorov, Y. Xiong, E. Chang, Y. Shi, R. Krishnamoorthi *et al.*, "Mobilellm: Optimizing sub-billion parameter language models for on-device use cases," in *Forty-first International Conference on Machine Learning*, 2024.

[17] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[18] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*. Springer, 2020, pp. 213–229.

[19] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[20] C.-Y. Wang, I.-H. Yeh, and H.-Y. Mark Liao, "Yolov9: Learning what you want to learn using programmable gradient information," in *European conference on computer vision*. Springer, 2024, pp. 1–21.

[21] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," *arXiv preprint physics/0004057*, 2000.

[22] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.

[23] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[24] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 4015–4026.

[25] H. Zhang, P. Zhang, X. Hu, Y.-C. Chen, L. Li, X. Dai, L. Wang, L. Yuan, J.-N. Hwang, and J. Gao, "Glipv2: Unifying localization and vision-language understanding," *Advances in Neural Information Processing Systems*, vol. 35, pp. 36 067–36 080, 2022.

[26] S. Harnad, "The symbol grounding problem," *Physica D: Nonlinear Phenomena*, vol. 42, no. 1-3, pp. 335–346, 1990.