Closed-Task Validation: A More Robust and Efficient Proxy for Guiding VLM Training

Enci Zhang[1,2] **Zongqiang Zhang**[1] **Jiahao Xie**[1] **Ruiqi Lu**[1]
Boyan Zhou[1] **Cheng Yang**[1 *]

eczhang@stu.pku.edu.cn
{zhangzongqiang, xiejiahao, luruiqi.v, zhouboyan}@bytedance.com
[1]ByteDance,
[2]Peking University

Abstract

Reliable and efficient validation is critical for guiding the resource-intensive process of training Vision-Language Models (VLMs). The standard evaluation paradigm, however, which relies on open-ended text generation, exhibits significant methodological limitations. We empirically demonstrate that this approach is unreliable, yielding high-variance metrics with a negligible correlation (r = 0.061) to final model performance. Furthermore, it is inefficient, as auto-regressive decoding introduces substantial latency and severe load-balancing issues in parallel evaluation. To address these limitations, we propose "Closed-Task" validation, a paradigm that bypasses auto-regressive decoding by converting questions into a multiple-choice format and directly inspecting token probabilities. Our experiments show this method is both highly reliable, producing stable signals strongly correlated (r = 0.798) with final performance, and efficient, achieving a >10x latency reduction with near-perfect load balancing. This work thus provides a robust and efficient validation methodology that resolves the interconnected challenges of evaluation reliability and system efficiency, offering a superior empirical framework for VLM development.

1 Introduction

Vision-Language Models (VLMs), such as GPT-4o [1] and Qwen-VL [2], have demonstrated remarkable capabilities, integrating deep visual perception with sophisticated natural language understanding. This fusion has unlocked unprecedented performance in complex multimodal reasoning, dialogue, and interaction [3–5], driving innovation across a diverse array of domains from robotics to accessibility tools [6, 7].

The process of training these sophisticated models, however, is extraordinarily resource-intensive, demanding massive-scale datasets—often containing millions of samples—and vast computational power, frequently consuming hundreds of thousands or even millions of GPU-days[8, 9]. Within this high-cost paradigm, continuous and reliable evaluation during training is not merely beneficial—it is operationally critical. Researchers and engineers depend on efficient validation proxies as an indispensable "compass" to monitor progress, select optimal checkpoints for deployment, and make crucial decisions on hyperparameter tuning, precisely because the full "gold standard" test sets are too large and expensive to run frequently[10]. An evaluation signal from such a proxy that is noisy or

[*] Corresponding author: yangcheng.iron@bytedance.com

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: .

misleading can lead to the premature termination of promising experiments or, conversely, the costly continuation of failing runs, resulting in a significant waste of resources[11].

Despite this necessity, the predominant evaluation methodology, which benchmarks performance on open-ended generative tasks like open-domain VQA, presents significant challenges to reliability. This unreliability manifests as high-variance, volatile evaluation curves (as shown in Figure 1), making it difficult to discern true model improvement from statistical noise. This volatility stems from several well-documented sources. 1) Metric-Induced Volatility: Traditional syntax-based metrics like Exact Match (EM) are "overly stringent"[12, 13], unfairly penalizing semantically correct paraphrases that deviate from a limited set of reference answers. 2) Data-Induced Volatility: VQA datasets inherently suffer from a "one-to-many" phenomenon, where a single question can have multiple valid answers due to ambiguity or subjectivity. A finite ground-truth set cannot capture this full space, penalizing novel yet correct responses[14]. 3) Inherent Instability: The generative process itself, including stochastic sampling and high sensitivity to training seeds, introduces significant randomness into evaluation scores[15].

Beyond its statistical unreliability, the open-ended paradigm suffers from a critical bottleneck in system efficiency. Open-ended evaluation is computationally expensive, dominated by the latency of auto-regressive decoding. This problem is exacerbated in the data-parallel settings required for rapid validation. As we will demonstrate (Figure 3), the long-tail distribution of output token lengths, often resulting from model "hallucinations" or repetitive outputs, leads to severe GPU load imbalance. This "straggler"[16] effect, where the entire batch must wait for the slowest GPU, results in dramatically reduced system throughput and wasted computational resources[17, 18].

To address this dual challenge of metric reliability and system efficiency, we propose a simple yet highly effective validation paradigm: Closed-Task Validation. Our approach leverages existing frameworks [19–21], to programmatically convert open-ended VQA validation tasks into a multiple-choice format. Critically, we bypass the generative process entirely. Instead of decoding a full answer, we prompt the model to select an option and evaluate its capability by directly inspecting the output probabilities of the single target tokens (e.g., 'A', 'B', 'C', 'D'), effectively eliminating the need for auto-regressive decoding.

Our experimental results, based on the training trajectory of a 'Qwen3-VL-4B' model, provide strong evidence for this paradigm shift. We demonstrate that:

- **Stability**: Closed-Task metrics (Figure 1) produce a stable, low-variance evaluation signal that clearly tracks the model's monotonic improvement, in stark contrast to the volatile and chaotic signal from Open-Task metrics.
- **Correlation**: The Closed-Task validation score exhibits a strong positive correlation ($r = 0.798$) with the model's "gold standard" performance on a comprehensive test set. Conversely, the Open-Task score shows no meaningful correlation ($r = 0.061$), rendering it an unreliable proxy for final performance (Figure 2).
- **Efficiency**: Our method yields a greater than 10x reduction in latency by eliminating the decode step. This, in turn, resolves the GPU load imbalance, enabling near-perfectly balanced and efficient parallel evaluation (Figure 3).

This paper introduces a new VLM evaluation paradigm that simultaneously solves the interconnected problems of metric reliability and system efficiency. We provide a practical, robust, and computationally efficient "compass" for VLM research and development, bridging the gap between volatile training signals and the need for dependable progress tracking.

2 Related Works

2.1 Reliability and Variance in Open-Ended VLM Evaluation

The evaluation of open-ended VLM generation is notoriously challenging, suffering from high variance that obscures true model progress[22, 10]. This instability arises from a confluence of factors[23, 24]. First, syntax-based metrics like Exact Match (EM) are "overly stringent"[12, 13], correlating poorly with human judgment by penalizing semantically correct paraphrases. Second, the datasets themselves are often ambiguous, featuring a "one-to-many" relationship where multiple valid answers exist for a single question[14]. A finite ground-truth set cannot capture this diversity, thus unfairly penalizing novel and accurate responses. Third, the training and inference processes are inherently stochastic. Sensitivity to training seeds, hyperparameter configurations, and non-deterministic sampling methods all introduce significant run-to-run variance[15]. While emerging LLM-as-judge evaluators[25] offer better semantic awareness, they introduce new challenges in computational cost and reproducibility. Our work bypasses this reliability crisis by reformulating the task to provide a stable, low-variance validation signal.

2.2 Efficiency and Systems for VLM Inference

Beyond reliability, the computational cost of open-ended evaluation presents a severe system-level bottleneck. The primary culprit is auto-regressive decoding, a sequential, memory-bandwidth-bound process that incurs high latency [26]. This latency problem is severely amplified in the parallel batch-processing setting required for rapid validation. The long-tail distribution of output token lengths—a common side-effect of model uncertainty or "hallucination"—creates a classic "straggler" problem in distributed systems [27, 28]. GPUs processing short-output tasks must idle, waiting for the one GPU handling a long-output "straggler,"[16] leading to catastrophic load imbalance and low resource utilization. A large body of research has focused on optimizing this bottleneck, with notable successes like the K/V cache management in vLLM [26] and I/O-aware kernels like FlashAttention [29]. These works, however, aim to mitigate the cost of decoding. Our work diverges by proposing an algorithm-system co-design: for the validation use-case, we eliminate the auto-regressive decoding process entirely, thereby sidestepping the root cause of both high latency and system imbalance.

3 Validation Methodology

3.1 General Experimental Setup

To empirically compare Open-task and Closed-task validation, we conducted a comprehensive Supervised Fine-Tuning (SFT) run. We trained a qwen3-VL-4B[30] model for 3100 steps on 16 A100 GPUs. The SFT data consisted of a curated mixture of over 5 million samples, blending public benchmarks with large-scale, application-specific data. For our "gold standard" evaluation, we used a large, held-out test set of 39.4k samples, composed of both proprietary, domain-specific tasks and public benchmarks. This large-scale test set is unsuitable for frequent validation for two critical reasons: 1) Using it for intermediate checkpoint selection would constitute data leakage, compromising its ability to measure true generalization. 2) It is computationally prohibitive, requiring two hours on 8 A100 GPUs for a single evaluation. Therefore, an efficient and reliable proxy is essential.

To rigorously evaluate the correlation of any validation proxy, it is necessary to compare it against the true model performance on our 'gold standard' test set. However, a full evaluation on this 39.4k-sample set is computationally intensive, making it infeasible to run at every validation step. Therefore, to construct a reliable ground-truth trajectory while balancing computational constraints, we adopted a principled sampling approach. Specifically, we selected key checkpoints every 300 training steps and performed a full evaluation on the entire test set for these points alone. This process generated the true performance curve depicted in green in Figure 1, providing a high-fidelity, albeit sparse, benchmark against which we could quantitatively measure the correlation of our validation proxies.

Our core experiment compares two distinct validation paradigms, which are applied to a custom-curated validation set described below.

3.2 Validation Set Curation and Annotation

Our validation set was constructed by drawing samples from six diverse benchmarks: ChartQA [31], InfoVQA [32], DocVQA [33], MathVista [34], MMVet [35], and VizWiz [36]. We utilized data from the AutoConverter [19] framework and its related dataset VMCBench, which provides open-ended questions and their corresponding multiple-choice (Closed-task) conversions. However, only a small subset of this data (e.g., 50 samples per dataset) contained ground-truth labels. To build a comprehensive validation set, we performed a new annotation step for the remaining samples. We employed an external large model, Gemini-2.5-Pro[37] and GPT-4o[1], to generate the ground-truth

Table 1: Composition of the 2,230-sample validation set, detailing the source datasets, sample counts, and the metrics used for Open-Task evaluation.

Dataset	Samples	Open-Task Metric	Closed-Task Metric
ChartQA[31]	450	Exact Match (Acc)	Exact Match (Acc)
InfoVQA[32]	450	ANLS	
DocVQA[33]	450	ANLS	
MathVista[34]	250	Exact Match (Acc)	
MMVet[35]	180	LLM-as-Judge	
VizWiz[36]	450	Exact Match (Acc)	

answers. To maximize annotation accuracy, we provided the annotator model with the full context, including both the original open-ended question and the context of the generated multiple-choice options. This process yielded our final 2,230-sample validation set, whose composition is detailed in Table 1.

3.3 Compared Validation Paradigms

Paradigm 1: Open-Task Validation (The Baseline). This represents the standard industry practice. For each validation sample, the model is prompted with the question and image, and it auto-regressively decodes a free-form text answer. This generated text is then evaluated against the ground-truth answers using task-specific metrics, such as Accuracy (Acc), Average Normalized Levenshtein Similarity (ANLS), or an LLM-as-judge score. As established in our related work, this paradigm is susceptible to both metric unreliability and system inefficiency.

Paradigm 2: Closed-Task Validation (Our Proposal). This is our proposed efficient and reliable alternative. This paradigm utilizes the multiple-choice conversions (e.g., A, B, C, D) of the validation questions. Crucially, we then bypass the auto-regressive decoding process entirely. We feed the model the image and the question, appended with the options, and constrain its output to a single token. We then directly inspect the model's output probability distribution over the option tokens ('A', 'B', 'C', 'D'). The option with the highest probability is selected as the model's answer. This approach eliminates generation, instead measuring the model's understanding and reasoning capabilities directly.

4 Validation Results and Analysis

4.1 Finding 1: Closed-Task Validation is a Stable and Highly Correlated Proxy

Our first major finding addresses the challenge of metric reliability. A useful validation proxy should provide a clear and interpretable signal of model progress, one that can be distinguished from statistical noise. Our experiments reveal a stark contrast in this regard: while the signal from Open-Task validation is often obscured by high variance, Closed-Task validation provides a far clearer and more stable indicator of model improvement. We establish this conclusion by evaluating both paradigms against two critical properties: stability and correlation.

The instability of Open-Task validation (red curves in Figure 1) is not an artifact of a single metric but a systemic issue rooted in the open-ended paradigm itself. The core challenge lies in the vast and unconstrained space of possible correct answers, which makes it exceedingly difficult for any metric to provide a stable signal. Our experiments highlight this difficulty across a spectrum of evaluation approaches. On datasets employing the stringent Exact Match (Acc) metric, such as ChartQA and VizWiz, the signal exhibits extreme oscillations, where minor syntactic variations lead to drastic score changes that could be misinterpreted as catastrophic forgetting. To mitigate this, one might turn to more semantically lenient metrics like ANLS. Yet, on InfoVQA and DocVQA, while the volatility is dampened, the signal remains fraught with high-frequency noise, still too erratic for reliable decision-making. Even the most advanced approach, using a powerful LLM-as-judge for MMVet, fails to resolve this fundamental issue. This progression of evidence compellingly demonstrates how the inherent variability of open-ended generation makes it exceptionally challenging to establish a stable and consistent evaluation signal. In stark contrast, our Closed-Task paradigm (blue curves) provides

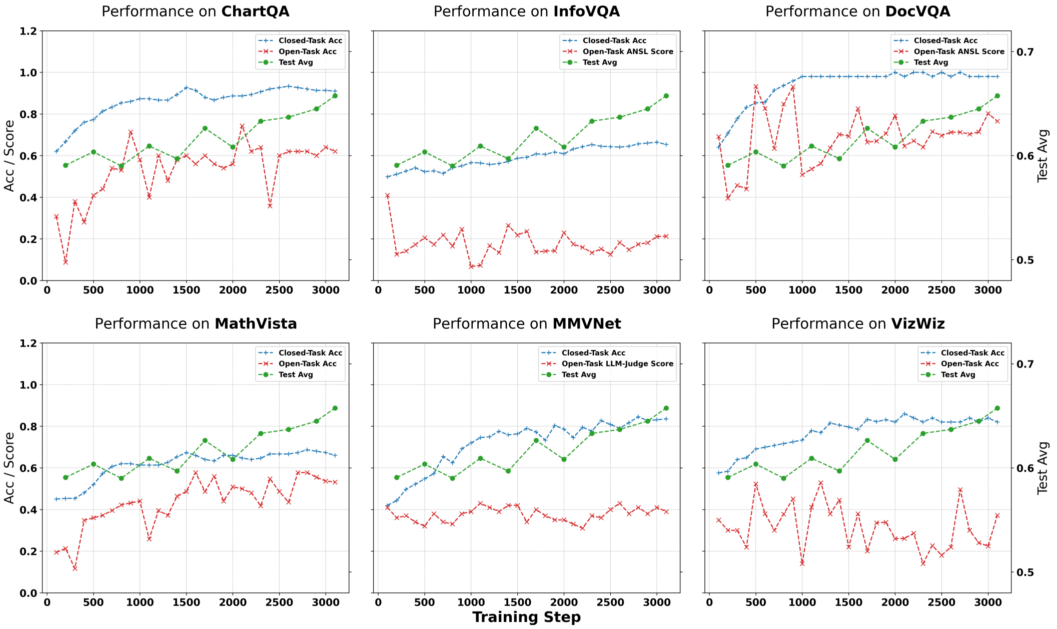

Figure 1: **Performance comparison of validation paradigms across six datasets.** The **blue curve (Closed-Task Acc)** and **red curve (Open-Task Acc)** correspond to the left Y-axis (Acc / Score). The **green curve (Test Avg)** represents the 'gold standard' performance on a large test set and corresponds to the right Y-axis (Test Avg) where applicable (e.g., in DocVQA and VizWiz). This figure demonstrates two key findings: (1) The Closed-Task signal (blue) is exceptionally stable and low-variance, while the Open-Task signal (red) is highly volatile. (2) The trend of the Closed-Task signal (blue) closely mirrors the trend of the true Test Average (green), proving it is a far more reliable proxy for model performance.

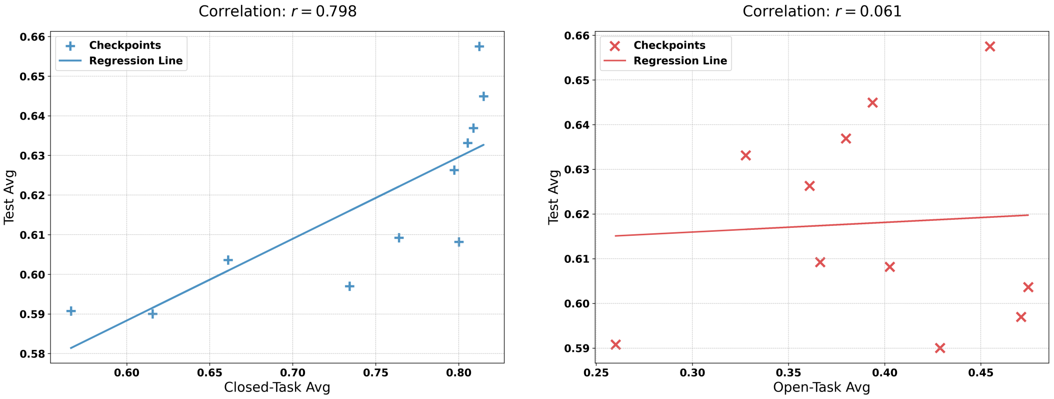

Figure 2: **Quantitative correlation analysis of validation metrics as predictors of final test performance.** Each point represents a training checkpoint, plotting its average validation score (X-axis) against its "gold standard" Test Average score (Y-axis). **(Left)** The Closed-Task validation score exhibits a strong positive correlation ($r = 0.798$) with the Test Average, as indicated by the tight clustering around the regression line. **(Right)** The Open-Task validation score shows no meaningful correlation ($r = 0.061$), with data points scattered randomly. This analysis provides statistical proof that Closed-Task validation is a highly reliable predictor of true model performance, whereas Open-Task validation is not.

a universally stable and interpretable signal. Across all six datasets, regardless of the underlying task complexity—from numerical reasoning in MathVista to chart parsing in ChartQA—the blue curve demonstrates a smooth, low-variance, and monotonic improvement. By reformulating the task to eliminate generative stochasticity, we successfully filter out the noise and isolate a clear signal of the model's evolving capabilities.

However, a stable signal alone is insufficient. It must also be a faithful proxy for the model's true generalization performance—our 'gold standard'. A metric that is stable but uncorrelated is just as misleading. A preliminary visual inspection in Figure 1 already offers compelling evidence. As described in our experimental setup (Section 3.1), we generated a "gold standard" performance trajectory (the green curve) by periodically evaluating key checkpoints on our full test set. On datasets where the 'gold standard' performance is plotted, such as DocVQA and VizWiz, the trend of our Closed-Task metric (blue) closely mirrors the true test average. Both curves capture the same learning dynamics, from rapid initial improvement to eventual convergence. The Open-Task metric (red), conversely, shows no discernible relationship to the ground truth. Figure 2 provides the definitive statistical verdict. The right panel demonstrates that the Open-Task validation score is a failed proxy. With a Pearson correlation coefficient of merely $r = 0.061$, the relationship between the metric and true performance is statistically negligible. The scattered data points confirm that this metric offers no predictive power. Conversely, the left panel confirms that our Closed-Task validation score is a highly reliable proxy. The strong positive correlation $r = 0.798$ and tightly clustered data points indicate that a higher score on our validation set is a robust predictor of superior performance on the final, held-out test set.

In conclusion, the Closed-Task paradigm successfully passes both critical tests of a reliable validation proxy: it provides a stable, low-noise signal of progress and is strongly correlated with true model performance. This finding empirically validates it as a statistically superior "compass" for VLM training, replacing a misleading and volatile indicator with a dependable and predictive one.

4.2 Finding 2: Closed-Task Validation is Orders-of-Magnitude More Efficient

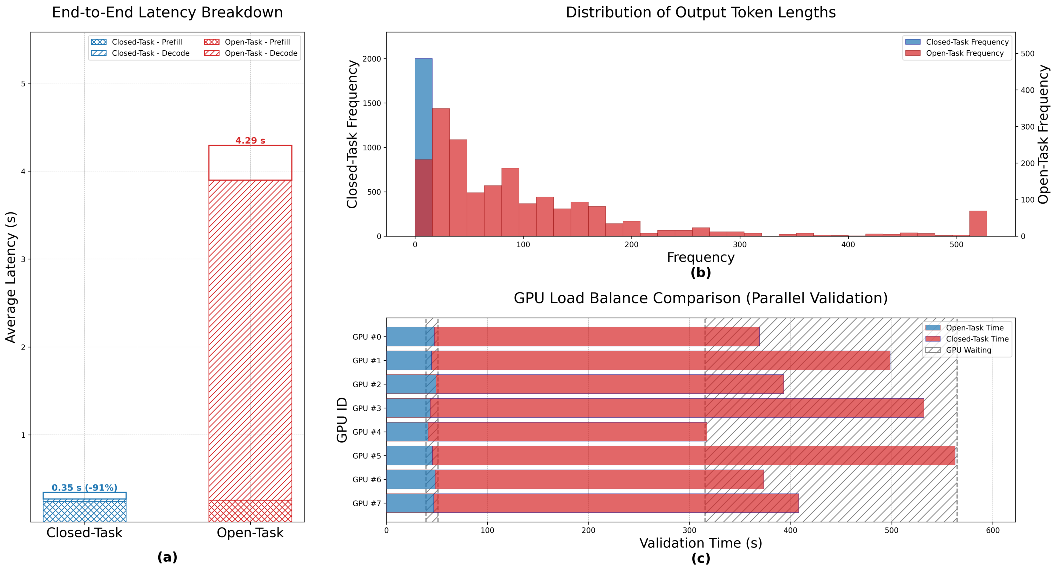

Figure 3: **Efficiency analysis of the Closed-Task versus Open-Task validation paradigms**. **(a)** Per-request latency breakdown, showing the Closed-Task paradigm achieves a 91% latency reduction by almost entirely eliminating the expensive auto-regressive Decode phase. **(b)** A histogram of output token lengths reveals the Open-Task paradigm suffers from a long-tail distribution—including "output collapse" (the spike at 512 tokens)—while the Closed-Task output length is fixed and predictable. **(c)** GPU load balance comparison in an 8-GPU parallel setting. The long tail from (b) causes a severe "straggler" problem for the Open-Task (red bars), leading to massive GPU Waiting (hatched) and low system utilization. In contrast, the Closed-Task (blue bars) achieves near-perfect load balance and maximum throughput.

Having established the statistical reliability of our paradigm (Finding 1), we now address the second critical challenge: system efficiency. We conducted a comprehensive and detailed empirical analysis of the validation performance characteristics of the two paradigms during the training process. The results are illustrated in Figure 3. The primary culprit for inefficiency in the standard paradigm is auto-regressive decoding, a sequential and memory-bandwidth-bound process. Figure 3(a) provides a stark quantification of this bottleneck. The Open-Task paradigm (Figure 3(a), red bar) incurs an average end-to-end latency of 4.29 seconds per sample. The latency breakdown definitively shows that the vast majority of this time is consumed by the `Open-Task - Decode` phase (red-hatched component). In stark contrast, our Closed-Task paradigm (blue bar) averages only 0.35 seconds. By reformulating the problem to a probabilistic check rather than a generative one, we almost entirely eliminate the auto-regressive decode step. This algorithmic shift yields a 91% reduction in latency, translating to a greater than 10x single-sample speedup.

his dramatic reduction in validation latency has profound practical implications. In a resource-intensive training environment, minimizing validation overhead directly increases the proportion of available GPU-hours dedicated to effective training. Furthermore, this efficiency gain allows for significantly more frequent validation intervals (e.g., validating every 100 steps instead of every 1000). This enables researchers to perform finer-grained monitoring of training dynamics and model convergence, allowing for earlier detection of issues like overfitting or divergence and providing a high-resolution signal for checkpoint selection.

However, the high average latency quantified in Figure 3(a) represents only one facet of the inefficiency. A more insidious system-level challenge stems from the stochastic variability of the decoding time, the statistical properties of which are revealed in Figure 3(b). This histogram plots the output token length distribution. The Open-Task distribution (Figure 3(b), red histogram) exhibits a pronounced long-tail. While many requests are of moderate length, a statistically significant portion requires extensive generation. This distribution is further characterized by an anomalous spike at the 512-token cutoff. This spike is a statistical artifact symptomatic of "output collapse," a known degenerate behavior in auto-regressive models, particularly prevalent during early training phases. This phenomenon manifests as the model regressing into a high-frequency pattern loop, failing to generate a coherent response or an End-of-Sequence (EOS) token. As a result, the decoding process does not terminate naturally and is instead truncated upon reaching the pre-configured `max_tokens` threshold, which directly explains the observed spike at that value. Conversely, the Closed-Task distribution (Figure 3(b), blue histogram) is deterministic and unimodal, consisting of a single mass point at 1 token. This stochastic, heavy-tailed workload characteristic of the Open-Task paradigm presents a fundamental obstacle to efficient parallel processing.

Figure 3(c) provides a direct visualization of this system-level consequence, often referred to as the 'straggler problem'[16] and this phenomenon is mathematically predictable by queueing theory [38]. In any data-parallel system where the total wall-clock time is defined by the $\max()$ of all worker completion times, a high-variance task distribution (as seen in Figure 3(b)) mathematically guarantees severe load imbalance. The total completion time is dominated not by the average-case, but by the statistically probable worst-case 'straggler' tasks drawn from the distribution's tail. This forces 'lucky' GPUs to remain idle, creating the significant `GPU Waiting` (hatched) area. Quantitatively, this imbalance is severe: for the Open-Task (red bars), the maximum GPU waiting time accounts for over 40% of the total wall-clock time. Conversely, the deterministic, zero-variance workload of our Closed-Task paradigm (Figure 3(b), blue histogram) eliminates this stochasticity. This results in near-perfect load balance and minimal idle time, as evidenced by the uniform blue bars in 3(c), where the maximum waiting time is kept well within 10% of the total runtime.

5 Limitations

While our findings present a strong case for the Closed-Task validation paradigm, we acknowledge several limitations that warrant consideration.

First, the efficacy of our paradigm is inherently linked to the quality of the multiple-choice conversion. The process of transforming open-ended questions into a high-quality, multiple-choice format with plausible distractors is non-trivial. Poorly constructed options could lead to an inaccurate or overly simplistic assessment of a model's capabilities. Although we utilized a powerful external model for this task, the methodology's performance remains dependent on the quality of this conversion step.

Second, our method is designed to evaluate a model's discriminative understanding and reasoning abilities—its capacity to identify the correct answer from a given set of options. However, it does not directly assess its generative fluency, creativity, or ability to formulate answers from scratch. A model could be proficient at selecting the correct option but still struggle with generating coherent, well-formed answers in a truly open-ended setting. Therefore, our validation proxy should be viewed as a powerful tool for tracking training progress, but not as a complete replacement for all forms of qualitative or generative evaluation, especially for final model assessment.

Finally, our empirical findings are based on the training trajectory of a single model architecture. While the underlying principles of our methodology are model-agnostic, further studies across diverse model families, scales, and training regimes would be necessary to fully establish the universal applicability of our conclusions.

6 Conclusion

This paper identifies key methodological limitations in standard open-ended VLM validation, highlighting its statistical unreliability from volatile metrics with negligible correlation to true performance, and its systemic inefficiency stemming from high latency and poor load balancing. Our proposed Closed-Task paradigm resolves these issues by bypassing auto-regressive decoding. This yields a stable, highly-correlated evaluation signal, an orders-of-magnitude efficiency gain, and near-perfect system utilization. Ultimately, this work provides a robust methodology that bridges the gap between rapid engineering iteration and reliable scientific measurement, offering a more effective 'compass' for future VLM research.

References

[1] OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, and Alex Baker-Whitcomb. Gpt-4o system card, 2024.

[2] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond, 2023.

[3] Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, Ruochen Xu, and Tiancheng Zhao. Vlm-r1: A stable and generalizable r1-style large vision-language model, 2025.

[4] Tao Gong, Chengqi Lyu, Shilong Zhang, Yudong Wang, Miao Zheng, Qian Zhao, Kuikun Liu, Wenwei Zhang, Ping Luo, and Kai Chen. Multimodal-gpt: A vision and language model for dialogue with humans, 2023.

[5] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks, 2019.

[6] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale, 2023.

[7] Yeseung Kim, Dohyun Kim, Jieun Choi, Jisang Park, Nayoung Oh, and Daehyung Park. A survey on integration of large language models with intelligent robots. *Intelligent Service Robotics*, 17(5):1091–1107, August 2024.

[8] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and Amy Yang. The llama 3 herd of models, 2024.

[9] Nestor Maslej, Loredana Fattorini, Raymond Perrault, Vanessa Parli, Anka Reuel, Erik Brynjolfsson, John Etchemendy, Katrina Ligett, Terah Lyons, James Manyika, Juan Carlos Niebles, Yoav Shoham, Russell Wald, and Jack Clark. Artificial intelligence index report 2024, 2024.

[10] Tony Lee, Haoqin Tu, Chi Heem Wong, Wenhao Zheng, Yiyang Zhou, Yifan Mai, Josselin Somerville Roberts, Michihiro Yasunaga, Huaxiu Yao, Cihang Xie, and Percy Liang. Vhelm: A holistic evaluation of vision language models, 2024.

[11] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. A survey on evaluation of large language models, 2023.

[12] Ehsan Kamalloo, Nouha Dziri, Charles L. A. Clarke, and Davood Rafiei. Evaluating open-domain question answering in the era of large language models, 2023.

[13] Chris Callison-Burch, Miles Osborne, and Philipp Koehn. Re-evaluation the role of bleu in machine translation research. 01 2006.

[14] Irina Saparina and Mirella Lapata. Ambrosia: A benchmark for parsing ambiguous questions into database queries, 2024.

[15] Gili Lior, Eliya Habba, Shahar Levy, Avi Caciularu, and Gabriel Stanovsky. Reliableeval: A recipe for stochastic llm evaluation via method of moments, 2025.

[16] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, January 2008.

[17] Zhewei Yao, Reza Yazdani Aminabadi, Olatunji Ruwase, Samyam Rajbhandari, Xiaoxia Wu, Ammar Ahmad Awan, Jeff Rasley, Minjia Zhang, Conglong Li, Connor Holmes, Zhongzhu Zhou, Michael Wyatt, Molly Smith, Lev Kurilenko, Heyang Qin, Masahiro Tanaka, Shuai Che, Shuaiwen Leon Song, and Yuxiong He. Deepspeed-chat: Easy, fast and affordable rlhf training of chatgpt-like models at all scales, 2023.

[18] Rui Zheng, Shihan Dou, Songyang Gao, Yuan Hua, Wei Shen, Binghai Wang, Yan Liu, Senjie Jin, Qin Liu, Yuhao Zhou, Limao Xiong, Lu Chen, Zhiheng Xi, Nuo Xu, Wenbin Lai, Minghao Zhu, Cheng Chang, Zhangyue Yin, Rongxiang Weng, Wensen Cheng, Haoran Huang, Tianxiang Sun, Hang Yan, Tao Gui, Qi Zhang, Xipeng Qiu, and Xuanjing Huang. Secrets of rlhf in large language models part i: Ppo, 2023.

[19] Yuhui Zhang, Yuchang Su, Yiming Liu, Xiaohan Wang, James Burgess, Elaine Sui, Chenyu Wang, Josiah Aklilu, Alejandro Lozano, Anjiang Wei, Ludwig Schmidt, and Serena Yeung-Levy. Automated generation of challenging multiple-choice questions for vision language model evaluation, 2025.

[20] Haohao Luo, Yang Deng, Ying Shen, See-Kiong Ng, and Tat-Seng Chua. Chain-of-exemplar: Enhancing distractor generation for multimodal educational question generation. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7978–7993, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[21] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark, 2024.

[22] Damien Teney, Peter Anderson, Xiaodong He, and Anton van den Hengel. Tips and tricks for visual question answering: Learnings from the 2017 challenge, 2017.

[23] Hugo Laurençon, Andrés Marafioti, Victor Sanh, and Léo Tronchon. Building and better understanding vision-language models: insights and future directions, 2024.

[24] Oscar Mañas, Benno Krojer, and Aishwarya Agrawal. Improving automatic vqa evaluation using large language models, 2024.

[25] Dongping Chen, Ruoxi Chen, Shilin Zhang, Yinuo Liu, Yaochen Wang, Huichi Zhou, Qihui Zhang, Yao Wan, Pan Zhou, and Lichao Sun. Mllm-as-a-judge: Assessing multimodal llm-as-a-judge with vision-language benchmark, 2024.

[26] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention, 2023.

[27] Tianyuan Wu, Wei Wang, Yinghao Yu, Siran Yang, Wenchao Wu, Qinkai Duan, Guodong Yang, Jiamang Wang, Lin Qu, and Liping Zhang. Falcon: Pinpointing and mitigating stragglers for large-scale hybrid-parallel training, 2024.

[28] Jinkun Lin, Ziheng Jiang, Zuquan Song, Sida Zhao, Menghan Yu, Zhanghan Wang, Chenyuan Wang, Zuocheng Shi, Xiang Shi, Wei Jia, Zherui Liu, Shuguang Wang, Haibin Lin, Xin Liu, Aurojit Panda, and Jinyang Li. Understanding stragglers in large model training using what-if analysis, 2025.

[29] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness, 2022.

[30] 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, 2025.

[31] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*, 2022.

[32] Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1697–1706, 2022.

[33] Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209, 2021.

[34] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023.

[35] Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*, 2023.

[36] Jeffrey P Bigham, Chandrika Jayant, Hanjie Ji, Greg Little, Andrew Miller, Robert C Miller, Robin Miller, Aubrey Tatarowicz, Brandyn White, Samual White, et al. Vizwiz: nearly real-time answers to visual questions. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 333–342, 2010.

[37] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, and Nathan Lintz. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities, 2025.

[38] Adam Wierman and Mor Harchol-Balter. Classifying scheduling policies with respect to unfairness in an m/gi/1. In *Proceedings of the 2003 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '03, page 238–249, New York, NY, USA, 2003. Association for Computing Machinery.