

Binary Verification for Zero-Shot Vision

Jeffrey Liu
mycube.tv

jeffreylieu@mycube.tv

Rongbin Hu
mycube.tv

rongbinhu@mycube.tv

Abstract

We propose a training-free, **binary verification** workflow for zero-shot vision with off-the-shelf VLMs. It comprises two steps: (i) quantization, which turns the open-ended query into a multiple-choice question (MCQ) with a small, explicit list of unambiguous candidates; and (ii) binarization, which asks one True/False question per candidate and resolves deterministically: if exactly one is True, select it; otherwise, revert to an MCQ over the remaining plausible candidates. We evaluate the workflow on referring expression grounding (REC), spatial reasoning (Spatial-Map, Spatial-Grid, Spatial-Maze), and BLINK-Jigsaw. Relative to answering open-ended queries directly, quantization to MCQ yields large gains, and True/False binarization provides a consistent additional boost. Across all tasks, the same workflow produces significant improvements, indicating generality. Our theory formalizes how open-ended vision queries can be quantized to MCQs and further binarized into True/False verifications, establishing a hardness ladder ($T/F \leq MCQ \leq K$ -way). A simple analysis explains why Boolean resolution boosts accuracy. Together, these components yield a simple and unified workflow that emphasizes inference-time design over task-specific training. It offers a practical, drop-in path to stronger zero-shot vision with today’s VLMs.

1. Introduction

Zero-shot vision [9, 36] is the ability of a model to perform a new visual task without any task-specific training or fine-tuning. It is increasingly critical for science and deployment. Compared with supervised learning, which requires task-specific annotations that are often expensive and time-consuming to build, zero-shot vision enables rapid and cost-effective adaptation to new tasks and domains. Moreover, proprietary vision language models (VLMs) often deliver stronger out-of-the-box performance and ship with mature APIs and infrastructure that simplify integration, but they typically restrict task-specific training beyond prompting, making inference-time methods especially valuable.

Modern VLMs [2, 3, 8, 12, 14, 20, 21, 26, 32] show strong zero-shot ability on tasks such as image captioning, broad scene recognition, and short-form visual QA. In typical zero-shot use, the model answers an *open-ended* prompt, for example, “Describe this image”. However, for more demanding tasks such as precise grounding, spatial reasoning, or fine-grained recognition, performance is often unreliable without task-specific fine-tuning. This unreliability stems from the interaction of several factors: free-form outputs make small numeric and referential errors both frequent and difficult to detect; model confidences are poorly calibrated, undermining decision thresholds; and more.

A fundamental idea in computing is that any number can be represented in a binary format. Quantizing and binarizing values to bits simplify arithmetic for machines and enable scalable computation. We adopt an analogous view and instantiate the idea of *quantization* and *binarization* for zero-shot vision tasks with a simple, training-free workflow. First, we *quantize* the task by generating candidate units, e.g. detector boxes, grid cells, or entity pairs, so an open query becomes a multiple choice question (MCQ) with a small set of explicit hypotheses. Next, we *binarize* the decision by asking, for each candidate h_i , a single yes/no claim that checks whether h_i matches the description, and the VLM must answer either True or False. Finally, we resolve outcomes deterministically: if exactly one candidate is True, we return it; if multiple answers are True, we restrict to that subset and issue a compact MCQ; if all are False, we fall back to the original MCQ.

We evaluate the unified workflow across three task families using the same VLMs without any task-specific fine-tuning, and observe significant improvements on every task.

- **Referring Expression Comprehension (REC).** REC [28] is the task of locating the specific object described by a natural-language expression in a given image. On REC (RefCOCO+/g) [24, 38], our binary verification workflow with GPT-4o reaches ACC@0.5 of 71.7%, 67.1%, and 64.8%, respectively, substantially above zero-shot GroundingDINO [22] (50.4%, 51.4%, 60.4%). Note that the original GPT-4o only has ACC@0.5 of 8.4%, 7.3%, 9.1%, respectively.

- **Spatial reasoning (Map, Grid, Maze).** For spatial reasoning [31], we evaluate three tasks, namely, Spatial-Map, Spatial-Grid, and Spatial-Maze, which probe directional relations, grid area recognition, and path finding, respectively. We take the original task interface as the baseline and then apply our two-step binary verification workflow. On **Spatial-Map**, the baseline is already MCQ. Binarization improves over MCQ from 78.3% to 84.5% and from 25.5% to 31.3% with GPT-4o [26] and CogVLM [32], respectively. On **Spatial-Grid**, spatial quantization, i.e., overlaying explicit grid on the original image, boosts the classification accuracy from 47.2% to 95.0% and from 23.2% to 58.8% with GPT-4o and CogVLM, respectively. On **Spatial-Maze**, spatial quantization by overlaying an explicit grid is again essential, which improves the accuracy of finding the right path from 16.5% to 68.7%. An embedding local T/F corridor checks further improves the accuracy to 75.7%.
- **BLINK-Jigsaw.** BLINK [10] is a multiple-choice benchmark that reformats classic vision tasks into image prompts to probe core perceptual skills that humans solve “in a blink,” but that current VLMs often miss. To test whether our approach helps on these harder perception cases, we evaluate a modified and harder BLINK-Jigsaw setting. Our method improves accuracy from 51.6% (MCQ) to 56.8%, demonstrating that binary verification still delivers measurable gains even when the underlying task is challenging for state-of-the-art VLMs.

Our approach is supported by a compact theory that mirrors the empirical trends. We show that any open-ended vision query can be *quantized* to a finite hypothesis set (an MCQ) and then *binarized* into per-option True/False checks, creating a hardness ladder under 0–1 loss: $R_K^* \geq R_m^* \geq R_2^*$. An information-theoretic perspective yields the same ordering, and when per-option verifiers are calibrated to posteriors, aggregating T/F answers recovers the Bayes K -way rule. A capability model further explains the gains: simpler prompts expose more task-relevant signal, tightening Fano bounds [7] on Bayes error for MCQ and especially for T/F. Finally, a two-hypothesis calculation shows that one-round binary verification typically exceeds MCQ.

This paper makes the following contributions:

- **Method.** We introduce a simple and unified workflow that *quantizes* an open-ended query to a small hypothesis set for a MCQ and then *binarizes* them into per-candidate True/False checks with deterministic resolution.
- **Experiment.** Using the same/similar workflow, we show broad gains across five tasks in three families, where quantization improves over open-ended prompting and binarization provides a further consistent boost.
- **Theory.** We develop a compact theory to justify the approach, clarify when and why it works, and support its generality across tasks.

2. Related Work

Inference-time workflows. Modern LLM/VLM usage increasingly treats generation as a *procedure* rather than a single shot. Chain-of-thought [34] elicits intermediate reasoning, ReAct [37] interleaves tool use with reasoning, and modular frameworks (e.g., DSPy [16]) compose these behaviors declaratively. Our work adopts this workflow view for vision, but focuses on a verification-centric interface.

Verification vs. selection. A large body of methods improves answers by *selection* over multiple candidates, including self-consistency/majority voting [33], process-of-elimination prompting [35], LLM-as-a-judge [23, 39]. These approaches remain MCQ-style, typically relying on sampling or confidence margins. In contrast, we convert each option into an independent True/False claim and apply a deterministic, codeword-style resolution over the boolean pattern.

Quantization via proposals. Zero-shot recognition and localization leverage vision–language pretraining (CLIP [29], ALIGN [15]) for open-vocabulary categorization, extend to localization with text-conditioned detectors (GLIP [19], OWL-ViT [25], GroundingDINO [22]), and expose generic masks via foundation segmentation models (SAM) [17]. We streamline zero-shot vision by a unified workflow.

Spatial reasoning benchmarks. Recent work shows that VLMs struggle on compositional and geometric reasoning even under MCQ interfaces. We adopt the NeurIPS 2024 Spatial-Reasoning suite (Map, Grid, Maze) [31] as a benchmark substrate and demonstrate that explicit spatial quantization by visible grids and indices, combined with binarization, substantially reduces error across all three tasks.

Our position. Relative to sampling-heavy selection and program-heavy tool pipelines, we propose a simple, unified, training-free workflow to streamline zero-shot vision tasks.

3. Methodology

Our method reframes prediction as verification through two transformations: *quantization* and *binarization*. Quantization converts an open query into a MCQ by proposing a compact, explicit set of hypotheses. Binarization converts that MCQ into a set of yes/no tests by rewriting each hypothesis as a precise claim. A VLM answers only True or False. Decisions are made by deterministic resolution over these binary outcomes.

3.1. Quantization

We first quantize the original task into a K -way discrete alphabet $\mathcal{A} = \{a_1, \dots, a_K\}$. For a given input, we then form a shortlist $S \subseteq \mathcal{A}$ with $|S| = m$ by proposing hypotheses that are *concrete and checkable*. Each hypothesis must be unambiguously referable in language (e.g., via coordinates, identifiers, or canonical labels). The design goal is

high coverage with parsimony: include the true state with high probability while keeping $|S|$ small enough to verify efficiently. After this step, the problem is an MCQ over the finite state space S .

In practice, quantization is *task-specific*. We quantize either the *answer space* (i.e., the admissible outputs) or the *search space* from which answers are derived. Concretely, (i) for object- or person-centric queries, use an open-vocabulary detector or tracker to produce a shortlist of regions/tracks with bounding boxes, masks, and/or IDs; (ii) for spatial queries, overlay an indexed grid to discretize the image plane, treating cells as candidates; (iii) for tasks with native discrete labels (e.g., expressions, attributes, compass directions), adopt the given label set as the quantized alphabet; (iv) when necessary, prompt a VLM/LLM to enumerate plausible answers under task-specific constraints. Overall, any proposal mechanism is admissible if it yields hypotheses that can be stated as precise claims about the input.

3.2. Binarization

Binarization replaces multi-way choice with a fixed set of independent True/False questions, one per MCQ option. For a m -option MCQ, we ask m separate questions of the form “*Is option i the correct answer?*” and require the VLM to return True or False to each question. The m answers form a boolean pattern over the option set, and the decision is made by deterministic rules that operate on this pattern:

- **Single-True.** If exactly one option is True, return it immediately.
- **Multiple-True.** If more than one option is True, treat this as verification noise and a useful constraint. Reduce the MCQ to the subset marked True and re-run per-option True/False on this reduced set. If the maximal allowed retries is reached without isolating a single True, construct a final MCQ containing only the True options and issue a single-shot MCQ prompt, and take that single-shot choice as the answer.
- **All-False.** If all options are False, regard this as under-specification. If retries remain, re-run binarization. If the maximal retries are reached, construct a final MCQ over all options and answer it in a single shot.

Concretely, for four options (a, b, c, d) , the pattern [True, False, False, False] selects a immediately, whereas [True, False, True, False] shrinks the MCQ to $\{a, c\}$ and triggers a refined two-way round. The deterministic resolution borrows the idea of *error detection* from channel coding [11]: per-option True/False outcomes act as a short codeword, enabling simple recovery without relying on probability margins. Binarization is largely *task-agnostic*. Any candidate set, however produced, is handled by the same per-option True/False queries with deterministic resolution.

Binary verification exposes a simple “*certainty knob*”:

instruct the verifier to treat uncertainty as False. This reduces false positives for Single-True and sharpens pruning when Multiple-True occurs. At the extreme, if the True criterion is so strict that every round is All-False, the scheme degenerates to the original single-shot MCQ. Hence the strictness of “True” can be the tuning parameter: looser thresholds may lead to a larger potential gain, while tighter thresholds favor a higher chance of non-negative gain.

3.3. Performance, Efficiency, Complexity

Performance. Binary verification improves accuracy through three mechanisms. First, quantization shrinks the search space by mapping an open-ended query to a small, explicit candidate set. Second, binarization reduces cross-candidate interference by localizing each decision to a single highlighted unit, limiting distraction when many proposals are present. Third, the deterministic resolution in binarization prunes the hypothesis set whenever single True or multiple True labels occur, improving the probability of a correct final choice.

Efficiency. Quantization controls cost by shrinking the shortlist of options of the MCQ. This yields predictable latency that scales with the shortlist size.

Complexity. We intentionally keep the resolution rule simple and practical. The T/F binarization already adds calls and therefore latency and cost. Piling on soft aggregation or heavier fusion schemes (e.g., noisy-OR [13] and BP/CRF pairwise consistency [18]) would further inflate compute and reduce reproducibility. In our deployment-oriented zero-shot setting, a deterministic, parameter-free rule is stable and compute-fair, while score-based or iterative alternatives add complexity for little gain.

4. Experiment

We study five vision tasks from three diverse families, namely REC [28], Spatial-Map, Spatial-Grid, Spatial-Maze, in spatial reasoning family [31], and BLINK-Jigsaw [10], with the single binary verification workflow. We evaluate the workflow mainly with a leading proprietary VLM (GPT-4o [26]) and a strong open-source VLM (CogVLM [32]), each representative of the state of the art in its category. In our experiments, we disable retries, where after quantization, we make a single pass of per-option True/False queries. If the outcome is not Single-True, we conclude with one final single-shot MCQ over the True subset if any, otherwise over all options.

4.1. Referring Expression Grounding (REC)

Task description. REC is the task of locating the specific object described by a natural language expression (e.g., ‘the red mug on the left’) in a given image. The standard output is a single bounding box that identifies the referent.

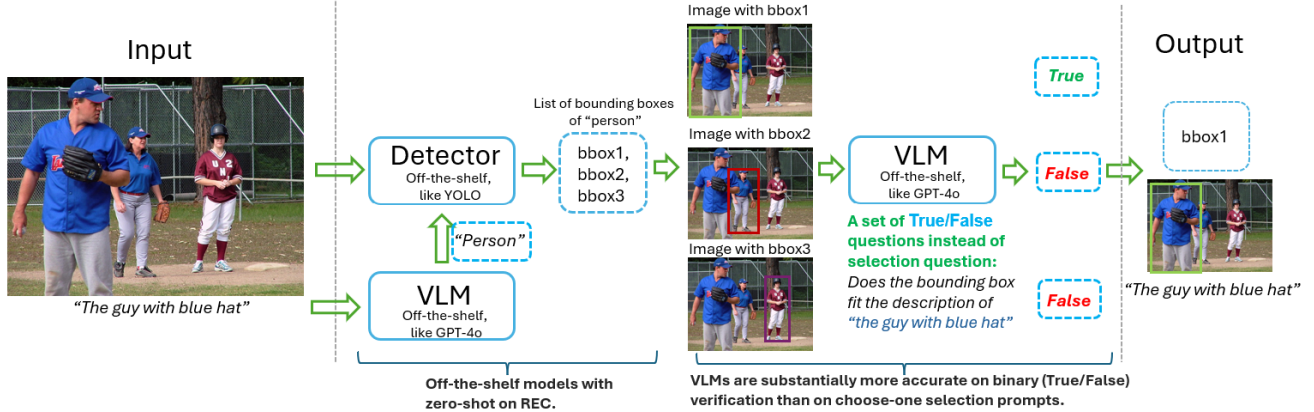


Figure 1. An example of binary verification workflow for REC.

Table 1. REC benchmarking (ACC@0.5, %).

Regime	Method	RefCOCO			RefCOCO+			RefCOCOg	
		Val	TestA	TestB	Val	TestA	TestB	Val	Test
Supervised	CogVLM (REC-trained)	92.6	94.3	91.5	85.2	89.6	79.8	88.7	89.4
Supervised	GroundingDINO (REC-trained)	–	77.3	72.5	–	72.0	59.3	–	66.3
Workflow with REC	GroundingDINO + CRG (REC-trained rerank)	–	81.6	73.2	–	77.0	60.0	–	69.6
Strict zero-shot	GroundingDINO (zero-shot baseline)	50.4	57.2	43.2	51.4	57.6	45.8	60.4	59.5
Zero-shot VLM (baseline)	GPT-4o (organic / vanilla selection)	8.4	–	–	7.3	–	–	9.1	–
Zero-shot on REC	MCQ (LLaVA, single-shot)	34.7	–	–	34.6	–	–	44.4	–
Zero-shot on REC	Binary verification (LLaVA)	44.6	–	–	42.3	–	–	50.7	–
Zero-shot on REC	MCQ (GPT-4o, single-shot)	62.1	65.6	58.4	59.6	61.0	53.7	62.2	61.0
Zero-shot on REC	MCQ (GPT-4o, majority voting)	63.6	–	–	60.2	–	–	62.6	–
Zero-shot on REC (ours)	Binary verification (GPT-4o)	71.7	77.8	63.0	67.1	70.1	58.5	64.8	63.7
Zero-shot on REC (ours)	Binary verification (GPT-5)	79.3	85.6	70.4	74.2	80.4	65.2	72.4	71.5

Table 2. RefCOCO (Val) binarization breakdown (GPT-4o).

	Single-True	Multiple-True	All-False
Frequency (%)	41.0	45.5	11.0
Candidate	3.6 → 1	7.0 → 4.2	3.5 → 3.5
ACC@0.5, %	80.8	71.7	54.8
ACC@0.5, %, MCQ	72.4	58.2	54.8

Quantization. As illustrated in Fig. 1, from the expression, we prompt the VLM to name a coarse object class c^* (e.g., *person*, *cat*), then run a class-conditioned detector on the image for c^* . We shortlist purely by detector confidence: retain only bounding boxes with score $\geq \tau$ and index them $\{a_1, \dots, a_m\}$, with m varying per image.

Binarization. For each shortlisted box a_i , we present the original image with bounding box i overlaid and pose the same binary question: “Does the object in the highlighted bounding box match the referring description? Answer with *True* or *False*.” The referring expression is appended verbatim. The VLM returns exactly *True* or *False*. After one pass over all boxes, we apply deterministic resolution: if exactly one box is *True*, we return it; if multiple boxes are *True*, we show only those boxes and issue a single-shot MCQ to select the best match; if all boxes are *False*, we

issue a single-shot MCQ over the original shortlisted boxes.

Results. We evaluate on RefCOCO, RefCOCO+, and RefCOCOg, i.e., the standard REC benchmarks derived from MS-COCO. A detection is correct if its Intersection over Union (IoU) with the ground-truth box exceeds 0.5. We therefore report ACC@0.5.

We compare three supervised REC baselines, including *CogVLM* (REC-trained VLM) [32], *GroundingDINO* [22] fine-tuned for REC, and a REC-trained workflow *GroundingDINO+CRG* that re-ranks detector proposals. Zero-shot baselines include *GroundingDINO* without REC fine-tuning and *GPT-4o* (vanilla REC) that directly outputs a box from a single prompt. Our zero-shot variants all use the same YOLO-World [6] proposals and differ in the VLMs and workflows. Note that YOLO-World is trained without COCO for dataset cleanliness. *Binary verification* (LLaVA¹/GPT-4o/GPT-5²) [21, 27] performs per-box True/False judgments with the deterministic resolution, while *MCQ* (LLaVA/GPT-4o) issues a single-shot MCQ over proposals. We also include *MCQ* (GPT-4o, majority

¹CogVLM includes supervised training for REC, so we instead use LLaVA-vicuna-13b as the representative open-source VLM.

²GPT-5 is included because REC has direct, real-world applications, so reporting the strongest available VLM is practically informative.

Table 3. Spatial-Map accuracy (%).

VLM	MCQ	MCQ ($5 \times MV$)	Binary verification
GPT-4o	78.3	78.7	84.5
CogVLM	25.5	25.7	31.3

voting) which repeats the single-shot MCQ three times and takes the vote. We use the default confidence level of 0.25 of YOLO-World, resulting in 5.02, 5.03, and 2.45 candidates on average for RefCOCO+/g(Val), respectively. We use temperature 0.2, while majority-vote runs use temperature 1.0, including the experiments in the following sections.

As shown in Table 1, our *binary verification (GPT-4o)* surpasses the zero-shot GroundingDINO baseline by a large margin. With GPT-5, binary verification further exceeds supervised methods with the sole exception of fully supervised CogVLM. Binary verification consistently outperforms MCQ, including MCQ with majority voting. Table 2 breaks down RefCOCO (Val) by boolean pattern, reporting ACC@0.5, frequency, and candidate counts pre/post pruning, showing where the gain comes from. The last row gives MCQ accuracy on the same items. 2.5% of items lack a valid candidate box. Note that the “organic” GPT-4o baseline attains less than 10% ACC@0.5 on these benchmarks, suggesting that it is not REC-trained.

4.2. Spatial-Map

Task description. The Spatial-Map [31] task presents a 2D map containing named locations and a natural-language question about a relative spatial relation between two targets, as shown in Fig. 2 (a). Spatial reasoning of this kind is known to be difficult for current VLMs due to sensitivity to absolute/relative spatial references.

Quantization. This task is natively an MCQ over a fixed set of directions, so we do not perform additional quantization.

Binarization. Given a question “Where is A relative to B?” with options {NW, SW, SE, NE}, we turn each option into a True/False claim and query the VLM once per option. The input is the map and the fixed claim, e.g., “A is northwest of B. Answer: True or False.” We issue the four binary queries individually and apply deterministic resolution.

Results. We evaluate GPT-4o and CogVLM, under three inference interfaces: (i) *MCQ* (single-shot choice among four directions), (ii) *MCQ ($5 \times MV$)* which repeats MCQ five times with fixed prompt and returns the majority vote, and (iii) *Binary verification*, our proposed binary True/False verification. Consistent with the REC findings, binary verification exceeds single-shot MCQ and MCQ majority voting. Table 3 summarizes results.

4.3. Spatial-Grid

Task description. The Spatial-Grid [31] task presents an image with an 5×5 visually discrete lattice and a natural-

language query that specifies a cell by row/column together with a multiple-choice list of categories (Fig. 2 (b)). The model must map the index to the correct grid cell and identify the object occupying that cell (e.g., *cat*). This formulation isolates two capabilities: (i) positional reasoning over a visually discrete lattice and (ii) in-cell recognition.

Quantization. The grid in the source image is only *visually* discrete, where it is likely that spatial reasoning conducted by a VLM is at continuous image level, rather than at cell level. We apply a *spatial quantization*: overlay visible grid lines (e.g., 5×5) so that the image is partitioned into disjoint regions with unambiguous cell boundaries and indices, as shown in Fig. 3a. This creates a two-dimensional quantization: (i) a *spatial alphabet* of cells defined by the overlay, and (ii) a *semantic alphabet* of animal categories provided by the options. The spatial quantization converts a fuzzy, pixel-level notion of “third row, third column” into a stable, language-aligned coordinate frame with unambiguous cell boundaries and indices. Note that the grid and overlay can be determined and generated by the VLM itself for a given image, rather than pre-defined rules.

Binarization. We verify *categories*, not cells. Given a query that specifies a target cell (r, c) on the overlaid grid, we show the *full* image with the grid and highlight (r, c) and issue one True/False claim per candidate category y : “The object in cell (r, c) is a $[y]$. Answer with True or False.” The category list is the MCQ option set. After a single round over all categories, we apply the deterministic resolution.

Results. We evaluate GPT-4o and CogVLM on Spatial-Grid under five inference schemes: (1) *MCQ (orig)* using the original image; (2) *MCQ (orig, $5 \times MV$)* repeating MCQ five times and taking the majority vote; (3) *MCQ (grid)* using the overlaid grid image; (4) *MCQ (grid, $5 \times MV$)* with majority vote; and (5) *Binary verification* using per-category True/False. As shown in Table 4, spatial quantization dramatically improves performance. Binarization further boosts the performance.

4.4. Spatial-Maze

Task description. The Spatial-Maze [31] task presents a 2D maze image with colored markers: the start cell **S** (green), the goal **E** (red), traversable corridor cells (blue) and walls (black). A natural language query asks for the path from **S** to **E** (Fig. 2 (c)). The path is reported in a set of critical point coordinates, including the starting point, all turning points, and the end point. Correctness is defined by the validity of the path.

Quantization. We apply spatial quantization by overlaying visible grid lines (e.g., 7×7) so the maze is partitioned into disjoint cells with boundaries and indices (Fig. 3a). The grid can be proposed by the VLM itself and rendered from its returned coordinates. Unlike Spatial-Grid, however, spa-

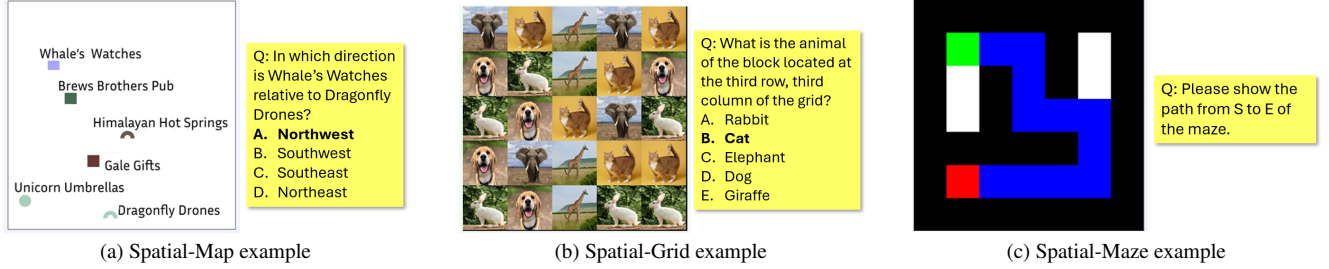


Figure 2. Illustrations for the Spatial-Reasoning family: for (a), we restrict to pairwise relations; for (b), we fix the query cell to (3,3), which is empirically harder; for (c), we directly request the full path (start, turns, end).

Table 4. Spatial-Grid accuracy (%). MV = $5 \times$ majority vote.

VLM	MCQ (orig)	MCQ (orig, MV)	MCQ (grid)	MCQ (grid, MV)	Binary verif.
GPT-4o	47.2	47.5	92.2	93.7	95.0
CogVLM	23.2	20.4	52.1	55.2	58.8

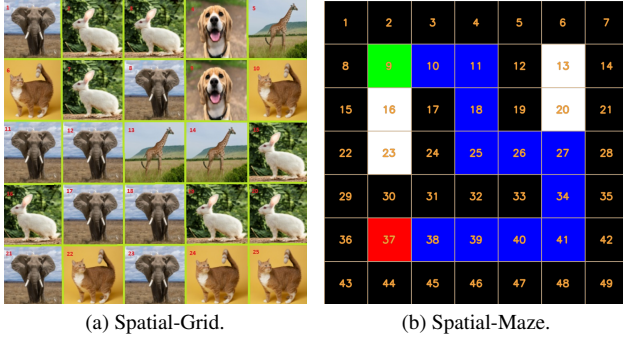


Figure 3. Explicit spatial quantization via visible grid overlays.

tial quantization does not generate a small semantic alphabet: even with a fixed grid, the target output is a *path*, and the induced K -way MCQ over all possible paths is combinatorial and potentially very large. Constructing a reliable shortlist that is both small and likely to contain the correct path is nearly equally difficult as constructing the right path.

Binarization. Spatial-Maze lacks a small MCQ shortlist, and we cannot apply the deterministic resolution on binary patterns proposed in Section 3. Instead, we embed binary verification inside the path-specification prompt. The model must output (i) an ordered list of indexes of the cells on the path, and (ii) a parallel list of per-step True/False checks of the form: “for each intermediate cell on the proposed path, is the cell blue?”. This variant preserves the principle of performing simpler verifications. However, it does not yield the same option-wise binarization in MCQ settings and the deterministic resolution.

Results. We evaluate Spatial-Maze with GPT-4o and CogVLM. A prediction is correct only if the entire route is valid, verified either from the coordinate set (start, turns, end) or from the sequence of cell indices. CogVLM almost never returns a valid path, suggesting that this task

Table 5. Spatial-Maze accuracy (%).

Method (GPT-4o)	Accuracy
No grid (original image)	16.5
Grid overlay	68.7
Grid overlay + T/F prompting	75.7

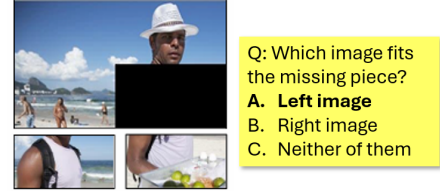


Figure 4. An example of BLINK-Jigsaw.

Table 6. BLINK-Jigsaw acc. (%) with 95% CIs (t).

Method (GPT-4o)	Acc.	95% CI
MCQ (Single-shot)	51.6	[49.6, 53.5]
MCQ (MV)	45.2	[43.9, 46.5]
Binary verification (Normal)	52.8	[51.4, 54.2]
Binary verification (Certainty)	56.8	[55.5, 58.0]

exceeds its current capability. We therefore omit its results. For GPT-4o, we report three settings: (i) *no grid* (original maze), (ii) *with grid overlay*, and (iii) *with grid + T/F prompting*. Accuracy improves with quantization dramatically and further with binary verification.

4.5. BLINK-Jigsaw

Task description. BLINK [10] is a multiple-choice benchmark that reformulates 14 classic vision tasks (e.g., relative depth, correspondence) into single/multi-image prompts to probe core visual perception that humans solve “in a blink,” but current VLMs often miss. As shown in Fig. 4, the BLINK-Jigsaw subtask is to select the tile that correctly

completes the image. To better reflect real-world deployments where a quantized option set may not cover the true answer, we extend the original two-way setup to a three-way MCQ by adding option (C) *Neither image is correct*. The “reject” choice also prevents a model from implicitly converting the task to a True/False decision (i.e., “*is A more likely than B to be the missing tile*”, rather than examining if A or B is indeed the missing tile. This increasing difficulty *significantly*. Compared with other BLINK sub-tasks, Jigsaw is more purely visual, less reasoning-heavy, and overlaps less with our other experiments, making it a complementary stress test.

Quantization. The BLINK-Jigsaw is a three-way MCQ.

Binarization. We follow the same binarization workflow by converting the MCQ into three per-option True/False queries, followed by the deterministic resolution. Because this task is especially challenging for current VLMs, we enable the certainty knob (Section 3.2) by instructing: “if uncertain, answer False” in each verification prompt.

Results. We report results on GPT-4o under four inference modes: (i) *MCQ*; (ii) *MCQ (3× MV)*, majority vote with three repeats; (iii) *Binary verification (normal)*, the True/False procedure without the “uncertain → False” policy; and (iv) *Binary verification (certainty policy)*, the True/False procedure with the “uncertain → False” policy. Although we refer to a *certainty knob*, current VLMs behave more like a *certainty switch*, where we cannot finely tune confidence and certainty, so the prompt enforces a conservative rule: if the model is not fully confident, answer False. Because the benchmark is small and only has 150 items, we evaluate each item eight times independently and aggregate the results, with confidence interval being reported. As this task appears to exceed CogVLM’s current capability, where its prediction is indistinguishable from random choice, we omit its results. As shown in Table 6, the binary verification with the “uncertain → False” policy introduces a clear improvement over single-shot MCQ; the normal binary verification may introduce a slight improvement; while 3× MCQ majority voting degrades accuracy. It is known that majority voting leads to lower accuracy than single-shot when voters are below chance and/or when errors are positively correlated.

4.6. Discussion

Across all five tasks, two consistent lessons emerge. First, quantization is critical. Moving from an open-ended prompt to a discrete MCQ formulation produces a dramatic jump in accuracy. The core reason is that, without quantization, the VLM may infer coordinates, relations, or fine-grained differences over a continuous image. After quantization, the model may only need to choose among a small number of well-defined candidates (e.g., grid cells and bounding

boxes). This reduces the visual search space and aligns language with a symbolic interface. The practical takeaway is simple: *if a vision question can be quantized into an explicit shortlist of candidate states, that should be done first*.

Second, binarization consistently provides an additional gain over MCQ. There are three main drivers. One is that answering a True/False question about a single candidate is usually easier than handling many candidates all at once. This is especially visible in REC, where MCQ requires showing multiple overlaid bounding boxes at once, which is visually noisy and induces cross-box interference, while the binary check inspects one highlighted box at a time, which is a simpler perceptual. The other driver is the deterministic resolution procedure, which plays a role similar to error detection in channel coding. Additionally, for challenging tasks, the certainty knob can be enabled to yield non-negative net gain. The lesson is that *our binarization scheme is an efficient alternative to majority voting when repeated queries are allowed within the latency budget*.

5. Theory

Problem formulation. A question $q \in \mathcal{Q}$ has answer space $\mathcal{Y}$ and a single correct answer $y^*(q) \in \mathcal{Y}$. Observed evidence is $X \in \mathcal{X}$ drawn from a task-dependent distribution $P_{X|y^*}$. *Quantization* is a measurable partition $\psi : \mathcal{Y} \rightarrow \mathcal{A}$ onto a finite alphabet $\mathcal{A} = \{a_1, \dots, a_K\}$. The discrete truth is $Y = \psi(y^*) \in \mathcal{A}$. A valid MCQ instance is a set $S \subseteq \mathcal{A}$ with $|S| = m$ and $Y \in S$ supplied with the question. *Binarization* queries, for each $a \in S$, the proposition “ $Y = a$ ” and receives an answer $b(a) \in \{\text{True}, \text{False}\}$. We write $\mathbf{b}_S$ for the boolean pattern on S .

Hardness ordering. The three tasks— K -way selection on $\mathcal{A}$, MCQ selection on a valid $S \subseteq \mathcal{A}$, and a binary test of a designated label—form a natural hierarchy. Under 0–1 loss, the optimal risks obey

$$R_K^* \geq R_m^* \geq R_2^* \quad (K \geq m \geq 2). \quad (1)$$

The argument is classical. Restricting an optimal K -way rule to any valid S cannot increase error, yielding the first inequality. Conversely, encoding an MCQ rule into a binary rule for a designated option (“is $Y = a$?”) cannot worsen performance, giving the second inequality. An information-theoretic view [5] reaches the same conclusion: each task identifies the cell of a partition of $\mathcal{A}$. Specifically, K -way uses K singletons, MCQ uses the m chosen options as singletons plus one merged “all others” cell, and a T/F check uses a binary split between one option and everything else. Since coarsening a partition cannot increase mutual information with X , we have

$$I(\Pi_K(X); Y) \geq I(\Pi_m(X); Y) \geq I(\Pi_2(X); Y), \quad (2)$$

Here $I(U; V)$ denotes the Shannon mutual information between random variables U and V :

$$I(U; V) = \mathbb{E} \left[\log \frac{p_{U,V}(U, V)}{p_U(U) p_V(V)} \right],$$

with the expectation taken under the joint law $p_{U,V}$.

A *partition* Π of the finite alphabet $\mathcal{A}$ induces a deterministic coarsening map $\pi : \mathcal{A} \rightarrow \{1, \dots, L\}$ that assigns each label to the index of its cell. Given the discrete truth $Y \in \mathcal{A}$, define the *coarsened label* $Z_\Pi \triangleq \pi(Y)$. By abuse of notation, $I(\Pi(\cdot); \cdot)$ in (2) should be read as $I(Z_\Pi; X)$. The three partitions used in (2) are:

- Π_K (finest, K cells): singletons $\{\{a\} : a \in \mathcal{A}\}$,
- Π_m (intermediate): the m MCQ options as singletons and the remainder merged into one cell, i.e., $\{\{a\} : a \in S\} \cup \{\mathcal{A} \setminus S\}$ with $|S| = m$,
- Π_2 (coarsest, binary): $\{\{a^\dagger\}, \mathcal{A} \setminus \{a^\dagger\}\}$ for a designated option $a^\dagger$.

With these definitions, (2) can equivalently be written as

$$I(Z_{\Pi_K}; X) \geq I(Z_{\Pi_m}; X) \geq I(Z_{\Pi_2}; X),$$

which expresses that coarser labelings (from K -way to MCQ to binary) cannot increase information about X .

Equivalence under calibrated posteriors. The hierarchy above does not preclude exact recovery of the K -way Bayes rule from per-option verifications when the latter are calibrated. Suppose a verifier returns

$$v(q, a) = \mathbb{P}[Y = a \mid X] \quad \text{for each } a \in \mathcal{A}. \quad (3)$$

Issuing the K per-option verifications and selecting

$$\hat{a}(X) \in \arg \max_{a \in \mathcal{A}} v(q, a) \quad (4)$$

produces the Bayes-optimal decision under 0–1 loss, because $\arg \max_a \mathbb{P}[Y = a \mid X]$ is the Bayes rule [4]. Restricting the argmax to a valid subset S yields the Bayes-optimal MCQ decision. Thus, in the ideal calibrated regime, one-vs-all verification is not merely a proxy for K -way classification. Instead, it is an equivalent realization.

Capability ordering. An information-budget view posits that the model exposes more task-relevant signal under simpler prompts [1, 30]. We write $t \in \{K, m, 2\}$ for the *prompt type* associated with a decision of cardinality $|S| = K$ (full K -way), $|S| = m$ (MCQ), or $|S| = 2$ (binary). The random variable $Y \in \mathcal{A}$ is the discrete ground-truth label after quantization. X denotes the observed evidence (e.g., image/text features) drawn from a task-dependent distribution. Given a fixed model and a chosen prompt type t , let O_t denote the model’s observable output under that prompt. We use $P_e^{(t)}$ for the *error rate* of the best achievable decision rule

under prompt type t for a task of size $|S|$. When applying Fano’s inequality, $P_e^{(K)}$, $P_e^{(m)}$, and $P_e^{(2)}$ correspond to $|S| = K, m, 2$, respectively.

With these definitions, the information-budget ordering and the induced Fano envelopes read:

$$I_K = I(Y; O_K \mid X) \leq I_m \leq I_2 = I(Y; O_2 \mid X), \quad (5)$$

$$P_e^{(K)} \geq 1 - \frac{I_K + \log 2}{\log K}, \quad (6)$$

$$P_e^{(m)} \geq 1 - \frac{I_m + \log 2}{\log m}, \quad (7)$$

$$P_e^{(2)} \geq 1 - \frac{I_2 + \log 2}{\log 2}. \quad (8)$$

Gain on boolean resolution. Consider a decision with two candidate hypotheses where exactly one is true. Let H_1 be the true hypothesis and H_2 a distractor.

A single-shot MCQ (i.e., one 2-way prompt) chooses H_1 with probability p , yielding

$$A_{\text{MCQ}} = p. \quad (9)$$

A per-option verifier answers True/False independently for each hypothesis. Let

$$q_1 \triangleq \mathbb{P}(\text{True} \mid H_1 \text{ is true}), \quad q_2 \triangleq \mathbb{P}(\text{True} \mid H_2 \text{ is false}). \quad (10)$$

If exactly one hypothesis is marked True, we accept it; if both are True or both False, we fall back to the same MCQ with success probability p . The resulting accuracy is

$$A_{\text{binary}} = q_1(1 - q_2) + q_1 q_2 p + (1 - q_1)(1 - q_2) p. \quad (11)$$

MCQ beats binarization iff $A_{\text{MCQ}} \geq A_{\text{binary}}$, which is equivalent to

$$p \geq 1 - \frac{q_2(1 - q_1)}{q_1(1 - q_2) + q_2(1 - q_1)}. \quad (12)$$

For any fixed $q_2 < \frac{1}{2}$, the right-hand side is increasing and concave in q_1 and lies strictly above the identity $p = q_1$ for $0 < q_1 < 1$. Thus, to match binarization, MCQ must be more reliable than the binarization on the true hypothesis (i.e., achieve $p > q_1$). Driving q_2 well below 0.5 via clearer claims or better rendering raises the bar further.

6. Conclusion

We propose a training-free, binary verification workflow that first *quantizes* an open-ended vision query into a small candidate set, then *binarizes* the decision into per-candidate True/False checks with deterministic resolution. Applied to five tasks across REC, spatial reasoning, and BLINK-Jigsaw, this workflow delivers consistent accuracy gains using the same off-the-shelf VLMs. Our analysis shows that these gains are theoretically supported.

References

- [1] Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. In *ICLR*, 2017. **8**
- [2] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, et al. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023. **1**
- [3] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. In *ICLR*, 2022. **1**
- [4] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, NY, USA, 2006. **8**
- [5] David Blackwell. Equivalent comparisons of experiments. *Ann. Math. Statist.*, 24(2):265–272, 1953. **7**
- [6] Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. YOLO-World: Real-time open-vocabulary object detection. In *CVPR*, 2024. **4**
- [7] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, Hoboken, NJ, USA, 2nd edition, 2006. **2**
- [8] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, et al. Instructblip: Towards general-purpose vision-language models with instruction tuning. In *NeurIPS*, 2023. **1**
- [9] Andrea Frome, Greg S. Corrado, Jon Shlens, et al. Devise: A deep visual-semantic embedding model. In *NeurIPS*, 2013. **1**
- [10] Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, et al. Blink: Multimodal large language models can see but not perceive, 2024. **2, 3, 6**
- [11] Richard W. Hamming. Error detecting and error correcting codes. *Bell System Technical Journal*, 29(2):147–160, 1950. **3**
- [12] Kaiming He, Xinlei Chen, Saining Xie, et al. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. **1**
- [13] David Heckerman. Causal independence for knowledge acquisition and inference. In *Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 122–127, 1993. **3**
- [14] Wenyi Hong, Weihang Wang, Qingsong Lv, Jiazhang Xu, Wenmeng Yu, et al. Cogagent: A vision language model for gui agents. In *CVPR*, 2024. **1**
- [15] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, et al. Scaling up visual and vision-language representation learning with noisy text supervision. In *Proc. ICML*, 2021. **2**
- [16] Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, et al. DSPy: Compiling declarative language model calls into state-of-the-art pipelines. In *ICLR*, 2024. **2**
- [17] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, et al. Segment anything. In *ICCV*, 2023. **2**
- [18] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected CRFs with gaussian edge potentials. In *NeurIPS*, 2011. **3**
- [19] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, et al. Grounded language-image pre-training. In *CVPR*, 2022. **2**
- [20] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge. *arXiv preprint arXiv:2401.02413*, 2024. **1**
- [21] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. **1, 4**
- [22] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, et al. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. In *ECCV*, 2024. **1, 2, 4**
- [23] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: Nlg evaluation using gpt-4 with better human alignment. 2023. **2**
- [24] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana-Maria Camburu, Alan Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, 2016. **1**
- [25] Matthias Minderer, Alexey Gritsenko, Austin Stone, et al. Simple open-vocabulary object detection with vision transformers. In *CVPR*, 2022. **2**
- [26] OpenAI. GPT-4o system card, 2024. Technical report. **1, 2, 3**
- [27] OpenAI. Gpt-5 system card. <https://cdn.openai.com/gpt-5-system-card.pdf>, aug 2025. **4**
- [28] Yanyuan Qiao, Chaorui Deng, and Qi Wu. Referring expression comprehension: A survey of methods and datasets. *IEEE TMM*, 23:4426–4440, 2021. **1, 3**
- [29] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, et al. Learning transferable visual models from natural language supervision. In *Proc. ICML*, 2021. **2**
- [30] Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method. In *Proc. Allerton Conf. Commun., Control, Computing*, 1999. **8**
- [31] Jiayu Wang, Yifei Ming, Zhenmei Shi, Vibhav Vineet, Xin Wang, Yixuan Li, and Neel Joshi. Is a picture worth a thousand words? delving into spatial reasoning for vision-language models. In *NeurIPS*, 2024. **2, 3, 5**
- [32] Weihang Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, et al. CogVLM: Visual expert for pretrained language models. In *NeurIPS*, 2024. **1, 2, 3, 4**
- [33] Xuezhi Wang, Jason Wei, Dale Schuurmans, et al. Self-consistency improves chain of thought reasoning in language models. In *ICLR*, 2023. **2**
- [34] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, et al. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022. **2**
- [35] Yixuan Weng, Minjun Zhu, Fei Xiao, Bin Li, Shizhu He, et al. Large language models are better reasoners with self-verification. 2023. **2**
- [36] Yongqin Xian, Christoph H. Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning — a comprehensive evaluation of the good, the bad and the ugly. *IEEE TPAMI*, 41(9):2251–2265, 2019. **1**
- [37] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, et al. React: Synergizing reasoning and acting in language models. In *ICLR*, 2023. **2**
- [38] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. Modeling context in referring expressions. In *ECCV*, 2016. **1**
- [39] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, et al. Judging llm-as-a-judge: MT-Bench and the Chatbot arena. In *NeurIPS*, 2023. **2**