Answering Cross-Dimensional Geometric Visual Questions by Multi-constraint Spatial Reasoning

Dongling Li, Qi Chen, Jianxing Yu*, Hanjiang Lai, Yanghui Rao, Wenqing Chen, Jian Yin
School of Artificial Intelligence, Sun Yat-sen University, Zhuhai, 519082, China
School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, 510006, China
Key Laboratory of Sustainable Tourism Smart Assessment Technology, Ministry of Culture and Tourism, Zhuhai, 519082, China
School of Software Engineering, Sun Yat-sen University, Zhuhai, 519082, China
{lidling3, chenq539, yujx26, laihanj3, raoyangh, chenwq95, issjyin}@mail.sysu.edu.cn

Abstract

This paper focuses on the task of answering complex visual questions that involve cross-dimensional (like 2D to 3D) spatial reasoning. This task (called *SpatialQA*) can enhance the machine's spatial cognitive abilities in "*plane representation - space reconstruction - semantic inference*", having great application value. Existing methods often only recognize 1-D visual objects and relations, but they lack the ability to represent in a cross-dimensional space and fail to grasp structured geometric knowledge such as face-face topology and texture details. That would cause problems such as texture misalignment and topological confusion, leading to error accumulation and incorrect answers. To address this problem, we propose a new method with good cross-dimensional reasoning capabilities. In detail, we first analyze the input diagram, capturing its relations in the 2D plane. To derive the topological relations in the 3D space, we employ a dual-channel augmentation technique to retrieve topological isomorphic examples and geometric rules, supplementing the missing but crucial reasoning clues. We then design a multi-perspective verifier to find the inconsistencies of the macroscopic outlines, eliminating incorrect options. Based on visual clues, we develop a question-guided detector to analyze the texture details and relations of each surface finely, capturing inconsistencies in a micro level. That can correct the reasoning bias to derive the right answer. Moreover, we create a large-scale dataset with 22,483 samples to conduct evaluations. The results show the effectiveness of our method.

1 Introduction

Spatial reasoning *VQA* is an important cognitive ability of AI, aiming to answer visual questions involving geometrical relations, such as position, direction, and distance among objects, etc. Most traditional work focuses on the questions with visual objects and relations in a 2D plane (Hudson and Manning, 2019), including the positions, sizes, and connections of objects, such as "*What is on the left of the beetle?*" on the image. These questions mostly can be solved by shallow visual matching. In addition to these simple cases, there are many questions that require deep geometric understanding that have not been explored. As shown in Fig.(1), the question involves mapping a 2D unfolded diagram into a 3D geometric structure, asking the image objects (e.g., a '*flower*' and a '*star*'), as well as their face-to-face topological (such as adjacent and opposite) relations and the texture directions. However, this cross-dimensional mapping is not trivial. In a 2D plane, the arrangement of faces is discrete and flat. When folded into a 3D structure, previously non-adjacent faces may become adjacent as distant boundaries merge together. This reconstruction of spatial relations caused by dimensional transformation lacks intuitive mapping patterns and is hard to learn. Moreover, the observation angles of the folded 3D objects are multi-directional and uncertain. Their projection features vary greatly. When the angle alters (such as ob-

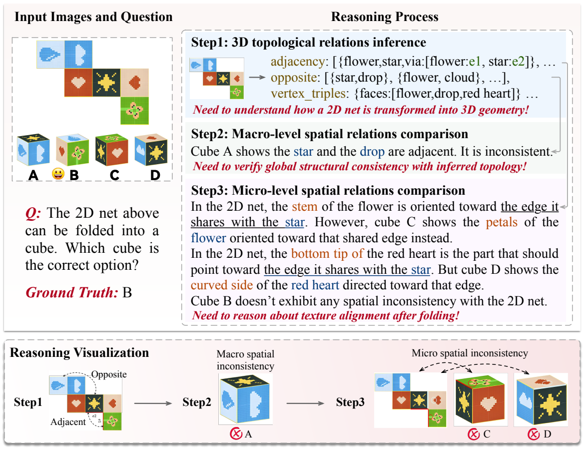


Figure 1: An example of the *SpatialQA* question.

* Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 33093–33111
July 2-7, 2026

serving from the front to the side or the top), the 3D-to-2D projection features and the visibility of the faces and textures will all change. It is hard to identify them via simple visual feature matching. This complex task has great value in applications such as industrial design, architectural engineering, and education. Less effort is devoted to exploring these complex questions. We thus propose a new task called *SpatialQA* to fill this research gap.

Early *VQA* methods mainly capture shallow visual features using neural networks (like *CNN*) to derive the answers. These matching-based methods lack the modeling of geometric topology. For example, it is unable to determine whether the non-adjacent 2D surfaces will be 3D adjacent or opposite after folding, failing to answer spatial questions. Another direction is the multimodal model, which uses an attention mechanism (like *Transformer*) to focus on the question-related visual features to improve the QA. However, this is still matching 2D shallow features with candidate 3D projection features, rather than modeling the mappings of 2D to 3D. It is susceptible to observation angle and has poor robustness. Some 3D modeling tools can reconstruct from 2D, but they are essentially passive geometric conversion tools and do not support QA. One solution is to resort to the relative invariance. That is, some geometric constraints that do not change with dimension transformation or observed angle, including topological invariance (such as the adjacency relations of 2D surfaces, the number of 2D surfaces, and the geometric shape of a surface) and texture invariance (like inherent textual content, which will not disappear or alter; the relative orientation of the texture within the same surface; and the geometric boundary of the surface) remain unchanged after 3D transformation. These verifiable constraints provide a good anchor for cross-dimensional reasoning, helping us derive the correct option via conditional consistency checks, but traditional methods fail to capture them to facilitate QA.

Based on the above observations, we propose a new cross-dimensional reasoning framework. We utilize relative invariance to filter out incorrect options that violate it. That can narrow the candidate range to reduce the reasoning complexity, locking in the correct answer that satisfies all constraints. In detail, we first analyze the input image to capture the objects and relations within the plane. Considering that 2D unfolded diagrams of the same shape may correspond to multiple 3D structures, relying solely on 2D features makes it hard to learn the 3D mapping. Thus, we exploit a dual-channel augmentation technique to retrieve topological isomorphic examples and geometric rules. They provide prior knowledge (such as adjacency/relative relations), as well as topological and texture constraints. By supplementing these missing but crucial reasoning clues, the model does not need to derive the 3D structure from scratch. It can obtain a batch of quantifiable judged conditions to determine the valid result, facilitating cross-dimensional spatial reasoning. We then design a macro verifier to eliminate incorrect options that clearly violate the topological invariance, avoiding the error accumulation problem. For example, if an option with two surfaces that should be adjacent in the 2D unfolded diagram are presented as opposite surfaces. That violates the adjacency invariance and can be filtered out. Moreover, we propose a question-guided detector that uses texture invariance to one-by-one locate the fine-grained inconsistencies like texture directions and positional relations. That can distinguish inherent attributes of texture and surface from projection deformation caused by different observation angles, thus enhancing the model's robustness and interpretability. In this coarse-to-fine progressive reasoning way, we can better answer complex spatial questions. Due to the lack of available datasets to evaluate our task, we construct a large-scale dataset named *SRQA*. We conduct extensive experiments on it and obtain significant improvement over other baselines.

The main contributions of this paper include:

- We propose a new task called *SpatialQA*. In contrast to traditional simple matching-based *VQA*, our task aims to answer complex geometric questions that involve cross-dimensional spatial reasoning over an image.
- We propose a multi-constraint framework with good spatial reasoning ability of topological and texture relations. It can well answer complex cross-dimensional spatial questions.
- We construct a large-scale evaluation dataset for this task to facilitate research in this field. Extensive experiments are conducted to examine the effectiveness of our approach fully.

2 Approach

This work studies a cross-dimensional spatial reasoning task. Given a question Q and visual inputs

$V = \{I_{2D}, \mathbf{I}_O\}$, where I_{2D} represents the 2D unfolding diagram and $\mathbf{I}_O = \{I_O^1, \ldots, I_O^k\}$ denotes the set of candidate images, the goal is to select the correct option from $\mathbf{I}_O$. As illustrated in Fig.(2), our method consists of three steps. First, we analyze the input image and deduce the spatial topological relations implied by it. Then, we examine the macro-level arrangement of surface outlines to filter out candidate options exhibiting inconsistent spatial configurations. For the remaining candidates, we perform micro-level spatial relation verification on surface textures to infer the final answer. Next, we elaborate on each part of our method.

2.1 Spatial Topological Relations Acquisition

To answer such spatial questions, we need to establish a 2D-to-3D mapping. That is influenced by multiple factors, such as the surface's size, shape, texture distribution, and topology. Due to the dynamic topological nature and the explosive growth of the 2D-to-3D space, it is hard to directly learn this implicit mapping from the data. To tackle this issue, we propose to retrieve relevant samples and geometric rules and view them as external prior knowledge to augment the model. This can greatly reduce the model's learning burden and avoid the inefficiency of zero-start derivation.

Topological Exemplar Retrieval. We observe that for a 2D diagram with a topological structure, its valid 3D folding results often follow certain patterns. That motivates us to retrieve topologically isomorphic samples and use them as a reference prior to facilitating the model learning of cross-dimensional mapping. This prior can also help the model transfer similar folding logic and avoid mapping failure in a few-shot distribution. That can enhance the model's generalization ability, but traditional methods (Shao et al., 2023) mainly rely on shallow visual feature matching (Zhang et al., 2024) rather than the intrinsic topological structure for retrieval (Schall et al., 2025). That would easily lead to the false retrieval of some samples that look similar from a certain observation angle but have different topological structures. Inspired by graph isomorphism theory and canonical labeling (McKay and Piperno, 2014), we propose a topology-aware retrieval algorithm based on graph canonization to tackle this issue. In detail, we first extract contours from the input 2D diagram and convert it into a binary image that preserves all face regions and their contact boundaries. This binary representation is then parsed into an undirected graph $G = (V, E)$, where the vertex set $V = \{v_1, \ldots, v_n\}$ denotes the face patches and the edge set $E \subseteq V \times V$ represents face adjacency relations. This graph is represented by an adjacency matrix $A \in \{0, 1\}^{n \times n}$. Next, we employ graph canonization to derive a unique adjacency matrix $A^* = P_{\pi^*}^{\top} A P_{\pi^*}$. It is invariant to node ordering. This guarantees that isomorphic graphs ($G_1 \cong G_2$) map to the identical representation. Building on this, we compute a topological hash $h(G) = \mathrm{Hash}(A^*)$ to query the pre-built sample pool $\mathcal{S}$, enabling the direct retrieval of the isomorphic sample s^*. Each sample consists of two parts: a 2D diagram and structural topological priors (such as face adjacency relations).

Related Rules Retrieval. Rules can effectively facilitate the reasoning of *MLLMs* (Huang et al., 2025), as predefined rules allow the model to execute condition matching and logical inference. Accordingly, we construct a structured rule base $\mathcal{R}$, where each rule $r \in \mathcal{R}$ is formalized as a tuple $r = \langle \mathit{Type}, \mathit{Condition}, \mathit{Conclusion}, \mathit{Keywords} \rangle$. Specifically, *Type* specifies the category of the polyhedron (e.g., hexahedron); *Condition* and *Conclusion* map 2D visual features directly to 3D spatial topological relations; and *Keywords* serve as semantic tags to assist retrieval. During the retrieval phase, we first employ *GPT-4o* to parse the input 2D diagram and extract geometric features (e.g., face count, layout structure), which we then use to formulate a standardized query q. Subsequently, we implement a '*filter-then-rank*' two-stage retrieval mechanism. First, irrelevant rules are rapidly filtered using *Type* as a hard constraint to obtain a candidate subset. Second, we employ a hybrid retrieval strategy to rank these candidates. For each query q and rule r, we compute the *BM25* keyword matching score S_{key} and the *Sentence-BERT* (Reimers and Gurevych, 2019) semantic similarity score S_{sem}. To address the scale discrepancy, both metrics are min-max normalized into $[0, 1]$. The final relevance score is determined by a weighted combination of these normalized terms, formulated as Eq.(1), where $\hat{S}_{(\cdot)}$ denotes the normalized score and α a hyperparameter. Finally, the top-k high-confidence rules are selected and injected into the model as context prompts to guide the reasoning process.

$$S(q, r) = \alpha \cdot \hat{S}_{\text{key}}(q, r) + (1 - \alpha) \cdot \hat{S}_{\text{sem}}(q, r) \tag{1}$$

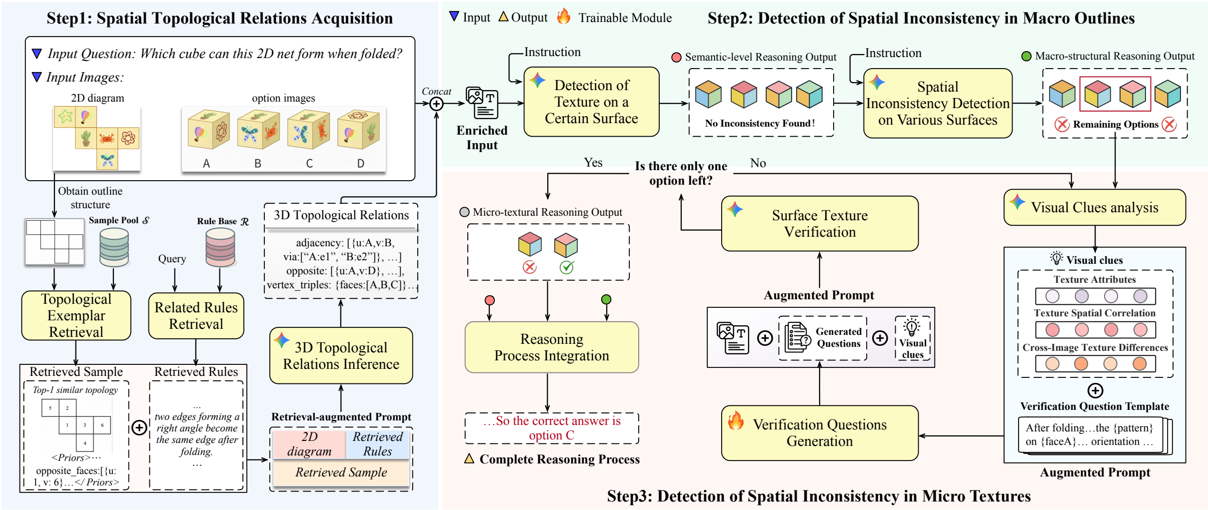


Figure 2: An overview of our proposed framework.

3D Topological Relations Inference. Based on the retrieved topologically isomorphic sample and relevant geometric rules, a retrieval-augmented prompt $\mathcal{P}$ra is constructed. It is then fed into an *MLLM* (i.e., *Gemini-2.5-Pro*) to infer the corresponding 3D topological relations $\mathcal{T}_{3D}$ of the 2D diagram. The inferred $\mathcal{T}_{3D}$ serve as explicit structural constraints for subsequent inconsistency detection across candidate options.

2.2 Detection of Macro Spatial Inconsistency

Instead of solving the entire problem in a single step, we decompose the complex spatial reasoning process into a sequence of inconsistency detection phases with progressively stronger constraints.

Texture on a Certain Surface. We scrutinize the semantic content of each candidate in $\mathbf{I}_O = \{I_O^1, \ldots, I_O^k\}$ to eliminate options containing texture features absent from the 2D diagram I_{2D}. For instance, a candidate showing an extraneous pattern (e.g., a bird not present in I_{2D}) cannot be folded from the given diagram and is therefore discarded. To perform this verification, we integrate the input question Q, input images (I_{2D} and $\mathbf{I}_O$), and 3D topological relations $\mathcal{T}_{3D}$ into an enriched input $\mathcal{E}_{\text{macro-1}}$, which is fed into an *MLLM* (i.e., *Gemini-2.5-Pro*) with the instruction $\mathcal{I}_{\text{tex}}$ to identify the invalid subset $\mathcal{O}_{\text{invalid\_tex}}$ that violates texture consistency. This process is formally defined as Eq.(2).

$$\mathcal{O}_{\text{invalid\_tex}} = f_{\text{MLLM}}(\mathcal{E}_{\text{macro-1}}, \mathcal{I}_{\text{tex}}) \qquad (2)$$

Inconsistency on Various Surfaces. Subsequently, we verify whether the relative positional relations among various surfaces in the 3D view adhere to the topological constraints derived from the 2D diagram. For instance, if a candidate depicts *Face A* and *Face C* as adjacent while the topology specifies that they are opposite, the candidate is marked as topologically inconsistent. Crucially, to ensure efficient reasoning, we update the candidate set by removing the options discarded in the previous step, yielding $\mathbf{I}'_O = \mathbf{I}_O \setminus \mathcal{O}_{\text{invalid\_tex}}$. We then construct a refined input $\mathcal{E}_{\text{macro-2}}$ based solely on these surviving candidates. This refined input is fed into an *MLLM* (i.e., *Gemini-2.5-Pro*) with the instruction $\mathcal{I}_{\text{topo}}$ to filter out the subset $\mathcal{O}_{\text{invalid\_topo}}$, as Eq.(3).

$$\mathcal{O}_{\text{invalid\_topo}} = f_{\text{MLLM}}(\mathcal{E}_{\text{macro-2}}, \mathcal{I}_{\text{topo}}) \qquad (3)$$

2.3 Detection of Micro Spatial Inconsistency

Following macro-level filtering, we examine micro-texture consistency, where existing *MLLMs* often struggle due to a tendency to rely on surface-level spatial relations and insufficient handling of rotation-invariant texture relations. Thus, we propose a question-guided verification mechanism. Since different texture types correspond to different spatial constraints, a unified verification strategy is often unreliable. To address this issue, we first introduce a visual clues analysis module that converts implicit visual information into explicit structured clues. Based on these clues, we train a generation model using reinforcement learning to adaptively construct discriminative verification questions. These questions are then provided as guiding prompts to a downstream verification model, directing its attention to key texture regions and relevant spatial relations.

Visual Clues Analysis. To provide reliable visual evidence for subsequent reasoning, we first conduct a fine-grained, multi-dimensional visual clues analysis on the remaining candidate images using an *MLLM*(i.e., *Gemini-2.5-Pro*). Specifically, we analyze the texture attributes on each visible surface of every candidate image, including the characterization of texture properties and their categorical types. In addition, we analyze the spatial correlations of textures across different surfaces, such as whether a texture is located near a shared edge or interacts with a shared vertex. Based on these analyses, we further identify the cross-image texture differences that are discriminative among candidate images.

Verification Questions Generation. To develop a verification strategy capable of precisely capturing fine-grained spatial discrepancies, we employ *Group Relative Policy Optimization* (*GRPO*) (Guo et al., 2025) to train the verification question generation policy. Specifically, at the decision step t, the environment state is defined as $s_t = \{\mathcal{O}_t, \mathcal{Y}_{\text{vis}}\}$, where $\mathcal{O}_t$ represents the set of candidate options currently remaining and $\mathcal{Y}_{\text{vis}}$ represents the structured visual clues. Conditioned on this state, the objective is to generate verification questions that specifically target micro-texture spatial consistency. We parameterize the policy π_θ using *Qwen2.5-7B-Instruct*. The agent's action is defined as generating a checklist containing multiple verification questions by adaptively selecting templates from a predefined set and filling their slots. Each template contains specific slots, which are populated with concrete information derived from the visual clues. Within the *GRPO* framework, for each input state s_t, the policy π_θ samples a group of G distinct checklists $\mathcal{A}_G = \{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_G\}$. Each $\mathbf{c}_g$ consists of M specific questions $\{c^{(1)}, \dots, c^{(M)}\}$.

To evaluate the effectiveness of each checklist, we design a composite reward function. For the g-th list $\mathbf{c}_g$ in the group, the comprehensive reward $R(\mathbf{c}_g)$ is calculated as Eq.(4).

$$
\begin{aligned}
R(\mathbf{c}_g) = \lambda_{\text{disc}}\, R_{\text{disc}}(\mathbf{c}_g) + \lambda_{\text{struct}}\, R_{\text{struct}}(\mathbf{c}_g) \\
+ \lambda_{\text{err}}\, R_{\text{err}}(\mathbf{c}_g)
\end{aligned}
\tag{4}
$$

Discriminative Reward (R_{disc}). To promote generating instance-specific verification questions, we design a reward based on the elimination rate of incorrect candidate options. Since higher-quality questions are more effective at pruning distractors, they thus yield higher rewards during policy optimization. Let $\mathcal{O}_t^{\text{inc}} \subset \mathcal{O}_t$ denote the set of incorrect candidates. The reward is defined as Eq.(5), where $\mathcal{O}_{\text{pruned}}(\mathbf{c}_g)$ represents the subset of incorrect candidates pruned based on verified inconsistencies.

$$
R_{\text{disc}}(\mathbf{c}_g) = \frac{|\mathcal{O}_{\text{pruned}}(\mathbf{c}_g)|}{|\mathcal{O}_t^{\text{inc}}|} \tag{5}
$$

Structural Base Reward (R_{struct}). To mitigate reward sparsity during early training and prevent the model from fixating on a single candidate, we introduce a structural base reward. This reward motivates the agent to generate checklists that fall within a valid length interval and encourage coverage over multiple remaining candidates. Let $\mathbb{L}_t = [|\mathcal{O}_t|, 2|\mathcal{O}_t|]$ denote the adaptive length interval derived from the number of candidate options $|\mathcal{O}_t|$, and let $\mathcal{T}(\mathbf{c}_g)$ denote the number of candidate options covered by the checklist $\mathbf{c}_g$. The structural reward is calculated as Eq.(6), where $\mathbb{I}(\cdot)$ is the indicator function.

$$
R_{\text{struct}}(\mathbf{c}_g) = \mathbb{I}\left(|\mathbf{c}_g| \in \mathbb{L}_t\right) \cdot \frac{|\mathcal{T}(\mathbf{c}_g)|}{|\mathcal{O}_t|} \tag{6}
$$

Error Penalty (R_{err}). To prevent irreversible errors during the elimination process, we impose a penalty when the checklist mistakenly removes the ground-truth option. Specifically, $R_{\text{err}} = -1$ if the correct option is excluded, and 0 otherwise.

After obtaining reward feedback, we compute the advantage via group-relative reward normalization to stabilize policy updates. For the g-th checklist $\mathbf{c}_g$, the advantage is computed as Eq.(7), where ϵ is a small constant ensuring numerical stability.

$$
\hat{A}_g = \frac{R(\mathbf{c}_g) - \text{mean}(\{R(\mathbf{c}_1), \dots, R(\mathbf{c}_G)\})}{\text{std}(\{R(\mathbf{c}_1), \dots, R(\mathbf{c}_G)\}) + \epsilon} \tag{7}
$$

The model parameters θ are updated by maximizing the objective in Eq. (8), where $\pi_\theta(a_t \mid s_t)$ denotes the current policy. A KL divergence term is applied to stabilize training against deviations from the reference policy.

$$
\mathcal{L}_{\text{GRPO}}(\theta) = \mathbb{E}\left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}\hat{A}_g - \beta \mathbb{D}_{\text{KL}}(\pi_\theta||\pi_{\text{ref}})\right] \tag{8}
$$

Surface Texture Verification. We concatenate the generated verification questions, visual clues, and remaining contextual information into a unified input, which is then fed into an *MLLM*(i.e., *Gemini-2.5-Pro*). Based on this input, the model identifies micro-level texture spatial inconsistencies between the input diagram and each candidate. If a single candidate remains after filtering, the pipeline proceeds to the subsequent reasoning fusion stage. Otherwise, the candidate set is updated and fed back to the visual clues analysis module, triggering a new round of checklist generation and verification. This iterative process continues until a unique candidate is determined.

To derive a final decision rationale that is both comprehensive and interpretable, we aggregate the reasoning outputs from the preceding stages (i.e., semantic-level $\mathcal{Y}_{\text{sem}}$, macro-structural $\mathcal{Y}_{\text{macro}}$, and micro-textural $\mathcal{Y}_{\text{micro}}$) into an integrated context $\mathcal{C}_{\text{integ}}$. Taking advantage of the strong contextual coherence capability of *MLLMs*, we feed $\mathcal{C}_{\text{integ}}$ into the model to synthesize these multi-stage reasoning processes into a logically coherent global reasoning chain $\mathcal{Y}_{\text{final}}$. This process is formally defined as Eq.(9), where $\mathcal{I}_{\text{fusion}}$ represents an instruction.

$$\mathcal{Y}_{\text{final}} = f_{\text{MLLM}}(\mathcal{C}_{\text{integ}}, \mathcal{I}_{\text{fusion}}) \tag{9}$$

3 Experiments

We extensively evaluated the effectiveness of our method with quantitative and qualitative analysis.

3.1 Dataset Construction and Settings

Existing benchmarks such as *SpatialViz* (Wang et al., 2025b), *MV-MATH* (Wang et al., 2025a), and *VisuLogic* (Xu et al., 2025) contain limited samples with spatial reasoning ability and lack sufficient diversity of surface textures. They are not suitable to evaluate our task. To tackle this issue, we constructed a large-scale dataset named *SRQA* via a three-stage pipeline. Step 1: *Data Acquisition*. We adopted two parallel streams. One was to utilize a crawler *Playwright* to harvest 1,986 raw samples from search engines. Another was to yield synthetic data via the 3D rendering tool *Blender*. The tool can model polyhedra well. Based on their 2D diagrams, it can map various *SVG* textures onto surfaces to enhance 3D visual diversity. Step 2: *Preprocessing*. All collected data underwent rigorous cleaning, including deduplication, label rectification, and format unification. Step 3: *QA Generation*. Based on the refined samples, we built multiple-choice geometric reasoning questions with one correct answer in *JSONL* format. We strictly adhere to data usage licenses and regulations. To ensure the quality, samples without ground-truth answers were subjected to a human-in-the-loop annotation protocol. Based on reasoning complexity, the questions can be divided into easy, medium, and hard levels. In this way, we obtained a large and diverse dataset with 22,483 samples. To avoid bias, we randomly partitioned our dataset into train, validation, and test sets in an 8:1:1 ratio. The set of statistics was given in Tab.(1) and more details are shown in Appendix A.

Classification	Easy	Medium	Hard	Total
Train	5,412	8,965	3,611	17,988
Val	683	1,109	453	2,245
Test	691	1,118	441	2,250

Table 1: Data distribution statistics of dataset *SRQA*.

We employed accuracy on multiple-choice questions as the evaluation metric. All *MLLM* baselines were evaluated using identical prompts and a temperature setting of 0.0. For closed-source models, we used the providers' API for inference. For the related rules retrieval module, the α in Eq.(1) was set to 0.6. Our experiments were on PyTorch (Paszke et al., 2019) with two NVIDIA A800 GPUs. During RL training, we set the group size $G = 8$ for group-relative reward normalization. The KL divergence coefficient β was set to 0.001. We optimized the policy using *AdamW* with a constant learning rate of 4×10^{-7} and a gradient clipping norm of 1.0. More details are provided in Appendix B.

3.2 Comparisons Against State-of-the-Arts

We conducted experiments on a broad set of multimodal large language models (*MLLMs*), including closed-source baselines: (1) *GPT-4o* (Hurst et al., 2024), *o1* (Jaech et al., 2024), *GPT-5*; (2) *Gemini-2.5-pro*, *Gemini-3.0-pro* (Comanici et al., 2025); (3) *Claude-3.7-sonnet* (Kurokawa et al., 2024). Open-source baselines included (4) *Qwen2.5-VL-7B-Instruct* and *Qwen2.5-VL-72B-Instruct* (Bai et al., 2025); (5) *InternVL3-8B* and *InternVL3-78B* (Zhu et al., 2025). (6) *Task Navigator* (Ma et al., 2024), a planning-based model that solved multimodal tasks by iteratively generating subgoals; and (7) *VoCoT* (Li et al., 2025), a chain-of-thought model designed for multi-step visual question answering.

As shown in Tab.(2), our model achieved the best performance across all difficulty levels, significantly outperforming both general *MLLMs* baselines and recent multi-step reasoning methods. Although *MLLMs* such as *GPT-5* and *Gemini-3.0-Pro*

Method	Easy	Medium	Hard	Overall
GPT-4o	30.5	23.3	14.5	23.8
o1	33.8	26.3	17.5	26.9
GPT-5	38.5	29.7	19.5	30.4
Claude-3.7-Sonnet	31.8	24.3	15.6	24.9
Gemini-2.5-Pro	37.8	29.3	19.5	30.0
Gemini-3.0-Pro	<u>43.0</u>	<u>32.2</u>	<u>20.9</u>	<u>33.3</u>
InternVL3-8B	25.8	20.1	11.8	20.2
InternVL3-78B	29.0	22.4	12.8	22.1
Qwen2.5-VL-7B†	27.2	20.2	12.4	20.8
Qwen2.5-VL-72B†	31.5	23.2	14.8	24.1
Task Navigator	34.3	26.0	16.4	26.8
VoCoT	34.9	26.3	17.1	27.1
Ours	**48.9**	**36.2**	**25.0**	**37.9**

Table 2: Main results on the *SpatialQA* task across various difficulty levels with a metric of accuracy (%). The best performance is **bold**, the second best is <u>underlined</u>. Models marked with † denote Instruct versions.

exhibited good visual-language alignment potential, their performance degraded substantially as task difficulty increased. That indicated they remained limited in reasoning about complex cross-dimensional geometric questions, especially in cases where there was spatial inconsistency at the micro-level. Besides, our model consistently surpassed the multi-step reasoning baselines, including *Task Navigator* and *VoCoT*. That suggested the task planning or visually grounded chain-of-thought reasoning alone was insufficient for handling cross-dimensional spatial reasoning. In contrast, our model addressed these challenges via coarse-to-fine spatial constraint verification. That can progressively eliminate structural topological conflicts and fine-grained texture orientation misalignments, providing a more effective solution.

3.3 Ablation Studies

To analyze the usefulness of our proposed components, we conducted ablation studies on four key parts of our model. (1) *w/o RAG*, removed the retrieval-augmented module and required the *MLLM* to infer the 3D topology directly without geometric prior knowledge. (2) *w/o VCA*, dropped the visual clues analysis module and generated verification questions only from templates and option images. (3) *w/o RL*, the reinforcement learning stage was removed, and verification questions were generated directly by a frozen *MLLM* without any policy optimization. (4) *w/o VQG*, discarded the verification questions generation module and let the model detect micro-level inconsistencies solely based on visual clues.

Mehod	Easy	Medium	Hard	Overall
w/o RAG	42.3	31.8	21.7	33.0
w/o VCA	44.2	33.4	22.8	34.7
w/o VQG	41.8	31.5	21.1	32.6
w/o RL	43.7	33.1	22.5	34.3

Table 3: Results of the ablation study.

As shown in Tab.(3), removing any component led to performance degradation of varying degrees. Among them, removing the question verification module caused the most significant drop. The overall accuracy of our model decreased from 37.9% to 32.6%, with a 3.9% reduction on the hard subset. That indicated the necessity of explicit verification questions to anchor the model's attention on critical texture regions. Removing reinforcement module also resulted in a consistent decrease, with the overall accuracy dropping by 3.6%. That suggested reinforcement learning could better train the model to yield more effective questions. Removing the visual clues analysis module further degraded performance, showing that this module helped our model interpret implicit texture details embedded in the image. Finally, removing the retrieval-augmented generation module caused a noticeable decline. That indicated external geometric priors were essential for deducing the correct 3D topology from the 2D diagram. Overall, these results showed that each module played a critical and complementary role in our model.

3.4 Evaluations on the Trade-off Parameters

Moreover, we evaluated the trade-off parameters in our reward function by univariate analysis. For each parameter, we tuned it in intervals of 0.2 over the range $[0, 1.0]$. When adjusting one parameter, the others were fixed at their empirically determined optimal values. Afterward, we plotted the performance curves in Fig.(3). We observed that the best result was achieved when the discriminative weight λ_{disc} was 1.0, with the structural weight $\lambda_{\text{struct}} = 0.2$ and error penalty $\lambda_{\text{err}} = 0.6$. That indicated the balanced strategy in terms of visual discrimination, structural integrity, and verification safety could achieve better performance.

3.5 Case Studies and Discussions

Furthermore, we conducted case studies to analyze the strengths of our method. As shown in Fig.(4), our model successfully detected orientation inconsistencies between surface patterns by

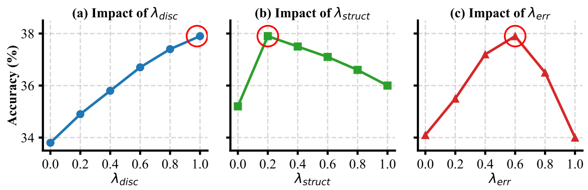


Figure 3: Evaluations of the trade-off parameters.

jointly leveraging explicit verification questions and the inferred 3D topological relations. In contrast, *Gemini-2.5-Pro* introduced reasoning errors due to intuition-based imagination and incorrect assumptions. *GPT-5* and *Qwen2.5-VL-72B-Instruct* tended to hypothesize a global folded configuration to guide their reasoning. Option selection was driven by matching high-level surface identities (e.g., front/top), without explicit verification of micro-level spatial relations. Moreover, their reasoning traces often exhibited unclear logic and omitted critical intermediate steps. As illustrated in Fig.(5), we provided a more detailed, step-by-step walkthrough of our model's inference process on a representative medium-difficulty example.

To gain deeper insights into our model's performance in complex scenarios, we conducted a detailed error analysis on the hard subset. Specifically, we randomly sampled 50 incorrectly predicted cases and analyzed their error patterns. We categorized the errors into four types, including (1) *Topology inference errors* (32%) occurred when the model failed to correctly derive the corresponding 3D topological relations from the 2D net. These errors typically manifested as correct local relations (e.g., opposite-face relations) but incorrect edge pairing or vertex aggregation. That would further affect subsequent texture-level reasoning. (2) *Texture-relation inference errors* (44%) arose when the model established 3D topology but misjudged texture spatial relations. That was due to the lack of clear directional or semantic clues in the texture. The model was hard to establish a stable reference frame for determining relative positions across surfaces. (3) *Visual perception errors* (18%) emerged from misidentifying key visual details in candidate options. They primarily occurred when different surfaces featured highly similar textures, leading to incorrect surface matching. (4) *Logical consistency errors* (6%), referring to failures when the inferred constraints were applied to the final decision. For example, the model might deduce correct intermediate texture relations but fail to reflect them in the answer. These challenges will be studied in future work.

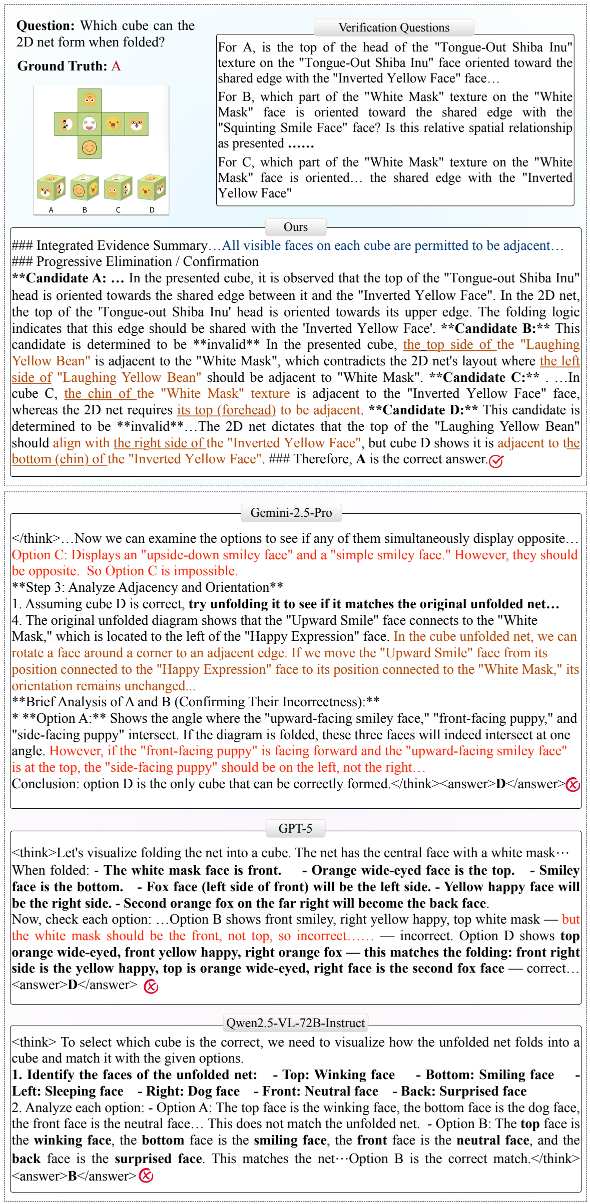


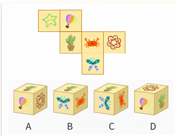
Figure 4: Case study. Red text highlights the erroneous reasoning steps.

4 Related Work

Spatial reasoning is a fundamental capability for *MLLMs* to perceive and reason about the physical world (Papadimitriou, 2025). Early research primarily focus on relative positional reasoning on the 2D plane (Kamath et al., 2023), such as inferring on-under or left-right relations (Hudson and Manning, 2019) from the observer's viewpoint (Liu et al., 2023). This type of spatial relation relies on shallow semantic matching signals and can be captured merely from 2D bounding boxes of the

Question: Which cube can the 2D net form when folded?

Ground Truth: C



Spatial Topological Relations

```
"adjacency": [{"u":"Cactus","v":"Hot Air Balloon","via":["top edge of Cactus", "bottom edge of Hot Air Balloon"]},…
"opposite": [{"u":"Cactus","v":"Flower"},{"u":"Hot Air Balloon","v":"Butterfly"},{"u":"Crab","v":"Star"}],
"vertex_triples": [{"faces":["Cactus", "Hot Air Balloon", "Star"]},…]
```

Verification Questions

"For B, which part of the 'Crab' on the 'Crab' face is oriented toward the shared edge with the 'Cactus' face? Is this relative spatial relationship as presented in the 2D net consistent?"

"For C, which part of the 'Butterfly' on the 'Butterfly' face is oriented toward the shared edge with the 'Cactus' face? Is this relative spatial relationship as presented in the 2D net consistent?"

Ours

```
### Integrated Evidence Summary
The analysis proceeds through a staged evaluation. An initial semantic check found no inconsistencies with the symbols used. The subsequent evaluation focuses on macro-structural
### Progressive Elimination / Confirmation
1. **Macro-structural Analysis (Face Adjacency):** * Candidates A and D are invalidated at this stage. The macro-structural reasoning (`Ymacro`) indicates that the Cactus face and the Flower face are opposite each other in the 3D structure derived from the net. Both candidates incorrectly depict these two faces as being adjacent, which is a topological impossibility.
2. **Micro-structural Analysis (Feature Orientation):**
* **Candidate B is invalidated. ** The micro-textural analysis reveals an inconsistency in the spatial relation between the Cactus and Crab faces. In candidate B, the 'cactus' side is at the top, and the 'crab' side is on the right. They share the bottom edge of the 'cactus' side and the top edge of the 'crab' side. The 'crab' portion facing this shared edge consists of its eyes and the top of its claws. However, In the 2D unfolded diagram, the "right" side of the cactus pot is along the 'cactus' side. The "left" side of the crab pot is along the 'crab' side, which is the right side of the crab's body and its legs (which appear to be the left side from the observer's perspective). When folded, the side of the pot should be adjacent to the crab's side legs.
* **Candidate C is validated. ** The micro-textural analysis (`Ymicro`) confirms that the orientation of the adjacent faces is correct. The analysis verified the relationship between the Cactus and Butterfly faces. In candidate C, the bottom edge of the Cactus face (the base of the pot) correctly adjoins the top part of the butterfly face (the butterfly's right wing), which is consistent with the spatial relationship established by folding the 2D net where the bottom edge of the Cactusmeets the left edge of the butterfly.
### Final Conclusion: Candidate C is the only option that is consistent with both the topological structure and the micro-textural orientation derived from the original net. Therefore, C is correct.
```

Gemini-2.5-Pro

```
<think>1. **Analyze the unfolded cube net:**The layout of the net is as follows…
2. **Determine the pairs of opposite faces:**
* In the horizontal strip `[Cactus] [Crab] [Flower]`, faces separated by one face are opposite. Therefore, the **Cactus** and the **Flower** are opposite faces.
* Let's visualize folding the net. If we consider the `[Crab]` face as the bottom of the cube:
* The `[Cactus]` folds up to become the back face.
* The `[Flower]` folds up to become the front face…
* From this folding, we can establish all the opposite pairs: Cactus is opposite to Flower. Crab is opposite to Balloon. Butterfly is opposite to Star.
3. **Eliminate impossible options:**
A fundamental rule of a cube view is that opposite faces can never be seen at the same time. We can use our list of opposite pairs to eliminate incorrect options.
* **Option A:** Shows the Cactus, Balloon, and Flower. Since the Cactus and the Flower are opposite faces, this cube is impossible.
* **Option D:** Shows the Flower, Balloon, and Cactus. Since the Cactus and the Flower are opposite faces, this cube is impossible.
4. **Check the picture orientations for this configuration:**
* **Cactus (Top):** In the net, the pot of the cactus points towards the `[Balloon]`. Since the `[Balloon]` is the back face, the pot should point towards the back edge of the cube. Both B and C show this correctly.
* **Crab (Right):** In the net, the top of the crab is adjacent to the `[Cactus]`. Since the `[Cactus]` is the top face, the crab's top should point upwards.
* In **Option B**, the crab is upright, with its top pointing up. This is correct.
* In **Option C**, the crab is tilted, with its top pointing towards the front-right corner. This is incorrect.
* **Butterfly (Front):** In the net, the top of the butterfly (its antennae) is adjacent to the `[Crab]`. Since the `[Crab]` is the right face, the butterfly's antennae should point to the right…
5. **Conclusion:** Based on the analysis of the picture orientations, Option B correctly represents the folded cube, while Option C has incorrect orientations for both the crab and the butterfly.
Therefore, B is the correct answer.</think> <answer>B</answer>
```

Figure 5: Case study. Red text highlights the erroneous reasoning steps.

objects. Subsequent studies expanded spatial reasoning to more diverse relation types, including distance comparison, size estimation, and multi-hop reasoning, accompanied by the development of more comprehensive benchmarks such as *SpatialVLM* (Chen et al., 2024), *Spatial-MM* (Shiri et al., 2024), *MMRel* (Nie et al., 2024), *BLINK* (Fu et al., 2024), and *SpatialRGPT*-bench (Cheng et al., 2024). However, these works primarily focus on recognizing single-dimensional visual objects and planar relations. In contrast, we target spatial reasoning in cross-dimensional scenarios, which remains a relatively underexplored area.

To enhance spatial relation reasoning, a variety of approaches have been proposed in recent years. Some studies (Cheng et al., 2024) formulated spatial reasoning as a structured modeling problem by constructing scene graphs or graph-based representations. That enabled fine-grained geometric inference over entities and relations (Yang et al., 2025). For more complex, multi-step spatial reasoning, several works introduced intermediate reasoning structures. For example, InSpire (Zhang et al., 2025) required models to generate explicit spatial reasoning steps, thereby reducing reliance on superficial appearance clues. Reinforcement learning has also been explored to improve robustness. Spatial-GRPO (Wang and Ling, 2025) perturbed spatial configurations to provide viewpoint-consistent reward signals, encouraging models to learn transformation-invariant geometric representations. Another line of work incorporates external knowledge via retrieval. Spatial-RAG (Yu et al., 2025) injected external geometric constraints via sparse retrieval to support spatial reasoning. While these diverse approaches have good spatial understanding within 2D or 3D scenes, they still struggle to solve cross-dimensional reasoning problems.

5 Conclusion

This paper investigated cross-dimensional spatial reasoning, a challenging task that required inferring latent 3D structures from 2D nets. We showed that existing *MLLMs* struggle with this problem due to unreliable inference of 3D topological relations and limited sensitivity to fine-grained spatial relations. To address the problem, we proposed a novel spatial reasoning framework driven by multi-level spatial inconsistency detections. We first employed a dual-channel retrieval strategy combining sample-based priors and geometric rules to deduce the latent 3D topological relations. We then applied macro-level verification to rule out global spatial inconsistencies. Finally, we introduced an active inquiry mechanism based on reinforcement learning. It can perform adaptive micro-level spatial verification via visual clues. Moreover, we constructed a large-scale dataset *SRQA*, and conducted extensive experiments on it. The results showed the effectiveness of our model. By using progressive spatial constraint checking and a question-guided texture detector, our model can achieve better robustness in complex cross-dimensional reasoning.

Acknowledgments

This work is supported by the Key-Area Research and Development Program of Guangdong Province (2026B0101100004, 2024B0101050005), National Natural Science Foundation of China (62276279, 62306344, U22B2060), Guangdong Basic and Applied Basic Research Foundation (2024B1515020032, 2024A1515010253, 2026A1515011800), Research Foundation of Science and Technology Plan Project of Guangzhou City (2023B01J0001, 2024B01W0004), Guangdong Philosophy and Social Sciences Planning Project (General Project Category) (GD24CGL57).

Limitations

This paper proposed a hierarchical inconsistency detection mechanism, which achieved effective performance improvements in complex geometric spatial relation reasoning tasks. However, this study still has several clear applicability limitations. First, this study focused on cross-dimensional tasks involving regular geometric shapes and did not cover samples with irregular surfaces or non-convex polyhedral structures. Second, this study derived 3D structures solely based on the topological and texture features of static 2D unfolded diagrams, without considering physical constraints (e.g., material toughness) or the influence of dynamic folding processes on spatial relations. Third, this study did not incorporate an interactive conversational module, nor did it design a model iteration strategy based on user feedback, and therefore did not support interactive spatial relation reasoning requirements. These aspects are not included in the current research scope of this paper and can be regarded as directions for future work.

Ethics Statement

This work introduces *SpatialQA*, a new task for cross-dimensional spatial reasoning. The proposed task and method present low ethical risk. Specifically, the data used in this study consists of a combination of synthetically generated geometric data and publicly available images collected from online sources. All collected data are used solely for research purposes and do not contain personal, sensitive, or identifiable information. Moreover, the task focuses on abstract geometric configurations, rather than social attributes or demographic categories, which reduces the risk of bias or misuse. In addition, our method emphasizes explicit reasoning and verification, aiming to improve model reliability and reduce hallucinated spatial inferences. As with any reasoning system, deployment in high-stakes scenarios should include appropriate safeguards and human oversight.

References

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. 2024. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14455–14465.

An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. 2024. Spatialrgpt: Grounded spatial reasoning in vision-language models. *Advances in Neural Information Processing Systems*, 37:135062–135093.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. 2014. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A Smith, Wei-Chiu Ma, and Ranjay Krishna. 2024. Blink: Multimodal large language models can see but not perceive. In *Proceedings of the European Conference on Computer Vision*, pages 148–166.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Ziyang Huang, Wangtao Sun, Jun Zhao, and Kang Liu. 2025. Improve rule retrieval and reasoning with self-induction and relevance reestimate. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5473–5488.

Drew A Hudson and Christopher D Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Amita Kamath, Jack Hessel, and Kai-Wei Chang. 2023. What's "up" with vision-language models? investigating their struggle with spatial reasoning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9161–9175.

Ryo Kurokawa, Yuji Ohizumi, Jun Kanzawa, Mariko Kurokawa, Yuki Sonoda, Yuta Nakamura, Takao Kiguchi, Wataru Gonoi, and Osamu Abe. 2024. Diagnostic performances of claude 3 opus and claude 3.5 sonnet from patient history and key images in radiology's "diagnosis please" cases. *Japanese journal of radiology*, 42(12):1399–1402.

Zejun Li, Ruipu Luo, Jiwen Zhang, Minghui Qiu, Xuan-Jing Huang, and Zhongyu Wei. 2025. Vocot: Unleashing visually grounded multi-step reasoning in large multi-modal models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3769–3798.

Fangyu Liu, Guy Emerson, and Nigel Collier. 2023. Visual spatial reasoning. *Transactions of the Association for Computational Linguistics*, 11:635–651.

Li Liu and Paul Fieguth. 2012. Texture classification from random features. *IEEE transactions on pattern analysis and machine intelligence*, 34(3):574–586.

Feipeng Ma, Yizhou Zhou, Yueyi Zhang, Siying Wu, Zheyu Zhang, Zilong He, Fengyun Rao, and Xiaoyan Sun. 2024. Task navigator: Decomposing complex tasks for multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2248–2257.

Brendan D McKay and Adolfo Piperno. 2014. Practical graph isomorphism, ii. *Journal of symbolic computation*, 60:94–112.

Jiahao Nie, Gongjie Zhang, Wenbin An, Yun Xing, Yap-Peng Tan, Alex C Kot, and Shijian Lu. 2024. Mmrel: Benchmarking relation understanding in multi-modal large language models. *arXiv preprint arXiv:2406.09121*.

Fivos Papadimitriou. 2025. Spatial ai for artificial general intelligence. In *Spatial Artificial Intelligence*, pages 53–63. Springer.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, and 1 others. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3982–3992.

Konstantin Schall, Kai Uwe Barthel, Nico Hezel, and Andre Moelle. 2025. A comprehensive approach to improving clip-based image retrieval while maintaining joint-embedding alignment. *Information Systems*, 134:102581.

Shihao Shao, Kaifeng Chen, Arjun Karpur, Qinghua Cui, André Araujo, and Bingyi Cao. 2023. Global features are all you need for image retrieval and reranking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11036–11046.

Fatemeh Shiri, Xiao-Yu Guo, Mona Golestan Far, Xin Yu, Reza Haf, and Yuan-Fang Li. 2024. An empirical analysis on spatial reasoning capabilities of large multimodal models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21440–21455.

Peijie Wang, Zhong-Zhi Li, Fei Yin, Dekang Ran, and Cheng-Lin Liu. 2025a. Mv-math: Evaluating multimodal math reasoning in multi-visual contexts. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19541–19551.

Peiyao Wang and Haibin Ling. 2025. Svqa-r1: Reinforcing spatial reasoning in mllms via view-consistent reward optimization. *arXiv preprint arXiv:2506.01371*.

Siting Wang, Luoyang Sun, Cheng Deng, Kun Shao, Minnan Pei, Zheng Tian, Haifeng Zhang, and Jun Wang. 2025b. Spatialviz-bench: Automatically generated spatial visualization reasoning tasks for mllms. *arXiv e-prints*, pages arXiv–2507.

Chenyun Wu, Mikayla Timm, and Subhransu Maji. 2020. Describing textures using natural language. In *Proceedings of the European Conference on Computer Vision*, pages 52–70.

Weiye Xu, Jiahao Wang, Weiyun Wang, Zhe Chen, Wengang Zhou, Aijun Yang, Lewei Lu, Houqiang Li, Xiaohua Wang, Xizhou Zhu, and 1 others. 2025. Visulogic: A benchmark for evaluating visual reasoning in multi-modal large language models. *arXiv preprint arXiv:2504.15279*.

Jihan Yang, Shusheng Yang, Anjali W Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. 2025. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10632–10643.

Dazhou Yu, Riyang Bao, Gengchen Mai, and Liang Zhao. 2025. Spatial-rag: Spatial retrieval augmented generation for real-world spatial reasoning questions. *arXiv preprint arXiv:2502.18470*.

Jianxing Yu, Qinliang Su, Xiaojun Quan, and Jian Yin. 2021. Multi-hop reasoning question generation and its application. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):725–740.

Jianxing Yu, Shiqi Wang, Libin Zheng, Qinliang Su, Wei Liu, Baoquan Zhao, and Jian Yin. 2023. Generating deep questions with commonsense reasoning ability from the text by disentangled adversarial inference. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 470–486.

Ji Zhang, Shihan Wu, Xu Luo, Hao Wu, Lianli Gao, Heng Tao Shen, and Jingkuan Song. 2025. Inspire: Vision-language-action models with intrinsic spatial reasoning. *arXiv preprint arXiv:2505.13888*.

Kai Zhang, Yi Luan, Hexiang Hu, Kenton Lee, Siyuan Qiao, Wenhu Chen, Yu Su, and Ming-Wei Chang. 2024. Magiclens: self-supervised image retrieval with open-ended instructions. In *Proceedings of the 41st International Conference on Machine Learning*, pages 59403–59420.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, and 1 others. 2025. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*.

A Data Construction Details

A.1 Details of Our *SRQA* Dataset

We constructed new data through a unified three-stage pipeline, strictly adhering to data usage licenses and regulations. (1) Data Acquisition: This stage involved two parallel streams. First, we utilized Playwright to harvest 1,986 raw samples from search engines using keywords like "*spatial reconstruction*" and "*polyhedral folding*". Second, we generated synthetic data via Blender. This involved modeling polyhedra (e.g., tetrahedra and hexahedra), generating their 2D nets, and mapping diverse SVG textures onto surfaces to enhance visual variety. (2) Preprocessing: All collected data underwent rigorous cleaning, including deduplication, label rectification, and format unification. Specifically, low-resolution web images were enhanced via super-resolution algorithms, while textual metadata underwent segmentation and semantic consistency checks to ensure strict text-image alignment. (3) QA Generation: Based on the refined samples, we constructed multiple-choice spatial reasoning questions with one correct answer in *JSONL* format. To ensure label accuracy, samples without ground-truth answers were subjected to a human-in-the-loop annotation protocol, where ten engineering graduate students verified answers via majority voting. Finally, questions were divided into easy, medium, and hard levels according to the proportion of difficult options in the candidate set.

To ensure a rigorous, unified, and reproducible difficulty distribution across heterogeneous data sources, we adopt an automated difficulty stratification protocol defined at the option level. Specifically, given a 2D net and a 3D candidate option, an option is regarded as *difficult* if the simultaneously visible faces in the 3D option are not all directly connected in the corresponding 2D net, where connection includes sharing a common edge or vertex. Following this option-level criterion, let a question contain N candidate options in total, among which H are identified as difficult options. A question is labeled as **easy** if $H = 0$, all options involve only locally adjacent faces without rotation or orientation ambiguity. It is labeled as **medium** if $0 < H \leq \lceil N/2 \rceil$, indicating that difficult options are present but do not dominate the candidate set. Finally, a question is labeled as **hard** if $H > \lceil N/2 \rceil$, where difficult options constitute the majority of the candidates.

To facilitate this process, all samples, whether synthetically generated or collected from the web, were first standardized into symbolic structural representations. For synthetic data, structural information was directly extracted from the rendering engine. For data collected from the web, we applied a lightweight hybrid verification pipeline, where *GPT-4o* proposed an initial structural interpretation, which was subsequently manually validated to ensure accuracy.

A.2 Sample Pool Construction

We first selected 2D nets with diverse topological structures from the image dataset. Through contour extraction and binarization, we obtained binary images that retained faces and boundaries, which were then parsed into undirected graphs and adjacency matrices representing faces (as vertices) and adjacency relationships (as edges). Subsequently, we applied visual annotations to the nets by automatically assigning a unique ID to each face.

Specifically, faces were identified from the parsed graph structure, and each face was assigned a deterministic index. The corresponding ID was then rendered at an interior anchor point of the face region, producing numbered 2D diagrams that explicitly exposed face-level topology for downstream processing. Based on the annotated nets, we employed a graph canonicalization algorithm to convert the adjacency matrices into lexicographically unique canonical matrices and generated topological hashes. These hashes were used as primary keys to construct an index, enabling efficient retrieval of topologically isomorphic samples. Each indexed sample consisted of a 2D diagram and its corresponding structural topological priors, such as face adjacency relations. Specifically, we utilized *GPT-4o* to generate structural topological priors conditioned on the annotated nets. The results generated by *GPT-4o* were subsequently reviewed and corrected through light human verification to ensure consistency with strict geometric constraints. After conducting sample-level isomorphism verification and optimization with scarce samples, the final sample pool was formed.

A.3 Rule Base Construction

We constructed a rule base $\mathcal{R}$ that encodes deterministic topological invariants of polyhedral nets, providing geometric rules for 3D topology inference. Each rule $r \in \mathcal{R}$ was represented as a tuple $r = \langle \mathit{Type}, \mathit{Condition}, \mathit{Conclusion}, \mathit{Keywords} \rangle$. These geometric rules were collected from publicly available educational and reference resources that document classical folding properties of regular polyhedra. Specifically, we crawled materials on polyhedral nets from geometry textbooks, instructional websites, and encyclopedic sources (e.g., Wikipedia pages on Platonic solids and polyhedral unfolding). These resources described deterministic topological properties such as opposite-face relations, face adjacency preservation, and vertex-coincidence constraints after folding. We employed *GPT-4o* to consolidate and normalize the crawled textual descriptions into a unified representation of geometric rules. The model was used solely to summarize recurring folding principles expressed in natural language and convert them into structured representations, formalized as subgraph-based conditions on the 2D face-adjacency graph together with their corresponding 3D topological conclusions. Representative examples of the constructed rules are shown in Tab.(4).

Type	Keywords	Condition	Conclusion
Hex	Linear, Row	Three faces in a continuous row (F_1,F_2,F_3).	F_1 and F_3 are Opposite.
Hex	Z-shape, Zigzag	Four faces form a 2×2 box with offsets.	Ends of ‘Z’ are Opposite.
Tet	Center	Central triangle surrounded by three others.	Meet at one top vertex.

Table 4: Snapshot of the constructed rule base $\mathcal{R}$. Note that **Hex** and **Tet** denote Hexahedron and Tetrahedron, respectively.

A.4 Details of Verification Questions Templates

Following classic studies on texture perception (Liu and Fieguth, 2012) and visual analysis (Cimpoi et al., 2014; Wu et al., 2020), we categorize the textures on candidate surfaces into five types. (1) *Asymmetric semantic patterns.* These textures have clear directionality, such as animal icons, arrows, or droplet shapes. Their spatial consistency mainly depends on whether the relative orientation is preserved after folding. (2) *Line-based patterns.* They consist of straight lines, polylines, or regular curves. Such patterns may extend across multiple faces or terminate at vertices. Their geometric meaning in 3D comes from line direction and endpoint position. Spatial reasoning requires checking whether lines connect properly along shared edges or meet accurately at the corresponding vertices. (3) *Periodic structural patterns.* These textures are formed by repeated units, such as stripes, grids, or dot arrays. They exhibit a stable main orientation and a consistent repetition phase. (4) *Symmetric patterns.* These textures contain center or axial symmetry. Their relative alignment with adjacent faces should remain valid after folding. (5) *Non-structural or random patterns.* Although these textures lack explicit directionality, their density distribution, color-block arrangement, or gradient trends should preserve reasonable correspondence across surfaces after folding. Based on this categorization, we derive a set of texture-aware verification question templates (Yu et al., 2021) for each texture type. Specifically, we collect the solution traces of approximately 500 representative training instances and prompt *GPT-4o* to summarize the common verification patterns and spatial invariants implicitly used during correct reasoning. These automati-

Name	Explanation	Example
Option	A candidate option representing one possible three-dimensional folding configuration derived from a 2D net.	Option A
Face	A single face of a polyhedral object on which patterns, lines, or textures are located.	face with an arrow
AdjacentFace	A face that shares a common edge with the current face.	face adjacent to the arrow face
Pattern	A semantic pattern located on a face, which may be asymmetric or symmetric and is used for spatial reasoning.	cat, dog, flower
Line	A linear texture element on a face, used to analyze geometric relations such as intersection or parallelism.	straight line
Curve	A curved texture element on a face, used to analyze bending direction and orientation.	curved arc
Polygon	A polygonal pattern on a face, typically used to verify edge alignment after folding.	square
TextureUnit	A basic repeating or atomic element forming a periodic or non-structured texture on a face.	dot, stripe

Table 5: Definitions of symbols used in verification question templates.

cally induced templates are then carefully reviewed and refined through human inspection to ensure correctness, coverage, and consistency (Yu et al., 2023). The resulting templates enable the model to generate discriminative spatial consistency queries that explicitly target the spatial invariants associated with different texture types. Representative verification question templates for different texture categories are illustrated in Fig.(6).

B Implementation Details

B.1 Settings of All Evaluated Methods

Settings of *MLLM* Baselines: All models were evaluated using identical prompts (as shown in Fig.(7)) and a temperature setting of 0. For open-source models, we deployed these models on A800 servers and used the officially provided code to load the pre-trained models for inference. For closed-

Asymmetric Semantic Patterns

- For {Option}, which part of {Pattern} on {Face} is closest to the shared edge with {AdjacentFace}? Is this relative spatial relationship as presented in the 2D net consistent?
- For {Option}, which part of {Pattern} on {Face} is oriented toward the shared edge with {AdjacentFace}? Is this relative spatial relationship as presented in the 2D net consistent?
- For {Option}, which part of {Pattern} on {Face} connects to which part of {Pattern} on {AdjacentFace}? Does this spatial relationship change after folding the 2D net?
- For {Option}, does {Pattern} on {Face} point toward the intersection of the three visible faces, or toward the intersection it shares with another face? Does this spatial relation change after folding the 2D net?

Linear Textures

- For {Option}, does {Line} on {Face} intersect with the shared edge between it and {AdjacentFace}? Is this relative spatial relationship as presented in the 2D net consistent?
- For {Option}, is {Line} on {Face} parallel to the shared edge between it and {AdjacentFace}? Is this relative spatial relationship as presented in the 2D net consistent?
- For {Option}, does the endpoint of {Line} on {Face} coincide with the vertex where the three visible faces intersect? Is this relative spatial relationship as presented in the 2D net consistent?
- For {Option}, does the curvature of {Curve} on {Face} bend toward the shared edge with {AdjacentFace}? Does this spatial relationship change after folding the 2D net?

Symmetric Pattern Textures

- For {Option}, what positional relationship (closer to or farther from) does {Pattern} on {Face} have relative to the shared edge with {AdjacentFace}? Is this relative spatial relationship as presented in the 2D net consistent?
- After folding, does a certain edge of {Polygon} on {FaceA} in {Option} correctly align with the edge of {FaceB} as indicated by the 2D net?

Periodic Structured Textures

- For {Option}, what spatial relationship does the primary arrangement direction of repeated {TextureUnit}s on {Face} have with the edges or vertices of other faces? Is this relative spatial relationship as presented in the 2D net consistent?
- For {Option}, what relative positional relationship does the distribution region of repeated {TextureUnit}s on {Face} have with other faces? Does this spatial relationship change after folding the 2D net?

Non-Structured Random Textures

- After folding, does the density region of {TextureUnit} distribution on {FaceA} (such as a high-density region located on the side of the shared edge between two faces) remain in the corresponding spatial orientation?
- For {Option}, what relative positional relationship do the color patches of {TextureUnit} on {FaceA} have compared to {FaceB}? Does this spatial relationship change after folding the 2D net?

Comparison

- Both {OptionX} and {OptionY} contain a surface with {PatternA} and a surface with {PatternB}, but the orientation of the patterns relative to the shared edge of the two surfaces differs. Compare them with the 2D net and determine which orientation is correct.
- Both {OptionX} and {OptionY} contain a surface with {PatternA} and a surface with {PatternB}, but the relative positions of the patterns differ. Compare them with the 2D net and determine which option is correct.

Others

- For {Option}, does the shadowed region on {FaceA} lie closer to or farther from {FaceB}? Does this spatial relationship change after folding the 2D net?

Figure 6: Representative templates.

source models, we used the providers' API interfaces for inference.

Settings of *Task Navigator*: we adopted *GPT-4o* as the multimodal large language model (*MLLM*) and use *GPT-4* as the language model (*LLM*). The task decomposition prompt P is as shown in Fig.(8 and the refinement prompt R is as shown in Fig.(9). Unless otherwise specified, all models were evaluated using their default inference configurations.

Settings of *VoCoT*: We utilized the version built upon the *Mistral-7B* backbone initialized with *CLIP-ViT-L/14*. During inference, *following the official VoCoT evaluation protocol*, we employed the trigger prompt "*Locate key objects and provide bounding boxes in your thoughts*" to activate the model's *RefBind* mechanism, enabling object-centric grounding by interleaving coordinate prediction and visual feature extraction within the reasoning chain. The input image resolution was set to 336^2, following the standard *VolCano* configuration, and the model was evaluated in a zero-shot manner on our dataset.

Settings of our model: During inference, our model followed a multi-step reasoning procedure based on the proposed framework. We employed *Gemini-2.5-Pro* to conduct inconsistency detection.

You are currently an excellent expert in spatial relation reasoning. You should first provide your step-by-step reasoning process enclosed in <think></think>, and then give a single final answer (A, B, C, D,…) enclosed in <answer></answer>.
Format:
<think> reasoning process </think>
<answer> final option </answer>
Question:
<question here>
Images:
<images here>

Figure 7: Unified Evaluation Prompt for Baseline *MLLMs*.

Decomposition Prompt
You are given a multimodal spatial reasoning question involving a 2D net and multiple 3D candidate options.
Your task is **not to solve the problem**, but to identify the **first concrete, answerable sub-question** that must be resolved using visual information.
The sub-question should:
•Be directly grounded in the given image(s),
•Be specific and operational (i.e., answerable by visual inspection or description),
•Avoid any inference about the final answer.
Output **only one sub-question**, phrased as a clear instruction.

Figure 8: *Task Navigator*: Visual Sub-question Decomposition Prompt.

Refinement Prompt
You are given:
(1) a previously generated sub-question, and
(2) the multimodal model's answer to that sub-question.
Your task is to determine whether the sub-question is **sufficiently specific and verifiable** for downstream spatial reasoning.
If the sub-question is vague, underspecified, or conflates multiple visual aspects, **refine it into a more precise version** that:
•Explicitly enumerates objects, options, or surfaces when applicable,
•Reduces ambiguity in visual reference or spatial scope,
•Remains grounded in observable visual evidence.
If refinement is needed, output the refined sub-question.
Otherwise, output the original sub-question unchanged.

Figure 9: *Task Navigator*: Visual Sub-question Refinement Prompt.

B.2 RL Training Details

We adopted *Group Relative Policy Optimization* (*GRPO*) to train the verification question generation policy. The policy network was initialized from *Qwen2.5-7B-Instruct* and optimized directly via reinforcement learning, without any preceding supervised fine-tuning stage.

During training, environment states were constructed from the *SRQA* training set. Each state s_t consisted of the current set of remaining candidate options and structured visual clues extracted from the input diagram. Conditioned on s_t, the policy generated a group of verification question checklists, whose effectiveness was evaluated by a frozen multimodal verifier. Specifically, we used *Qwen2.5-VL-72B-Instruct* as the verifier via API-based inference, which was queried using a fixed prompt with greedy decoding (temperature $= 0$) to provide deterministic and reproducible reward signals. The verifier was used exclusively for reward computation during training.

We set the group size to $G = 8$, meaning that for each state s_t, the policy samples eight distinct verification checklists in parallel to enable group-relative reward normalization. The KL divergence coefficient was set to $\beta = 0.001$, with the reference policy π_{ref} fixed as the initial *Qwen2.5-7B-Instruct* model.

At each optimization step, we sampled 512 environment states, resulting in $512 \times 8 = 4096$ trajectories per rollout. Policy optimization was performed using AdamW with a constant learning rate of 4×10^{-7}. Gradients were accumulated to form an effective optimization batch of 128 states. All experiments were conducted using bf16 precision with gradient clipping set to 1.0. Training was performed for 400 optimization steps on 2 NVIDIA A800 GPUs. The training process is completed within a few hours.

B.3 Methods for Final Answer Extraction

To extract the final answer, we employed a rule-based extraction scheme. Specifically, when the model's response adheres to the predefined instruction, we directly extracted the content within the <answer></answer> tag using regular expressions. For other cases, we located the answer by matching various identifiers, including '<answer>', 'Answer:', 'Final answer', 'The answer is', 'The correct answer is', and 'Correct answer'. The final correctness is determined by whether the extracted result contains an uppercase option letter (A-F) that matches the ground truth. This strategy ensures consistent evaluation across diverse response formats.

C Prompt Design for Our Proposed Method

You are a spatial reasoning expert. Follow the instructions below to infer the 3D structure formed by folding a given 2D net (image1).

[Task Instruction]

Given a 2D net, infer its 3D topological relations in the following order:

1.Identify and explicitly name all faces using intuitive and descriptive labels;

2.Analyze adjacency relations between neighboring faces based on shared edges;

3.Determine opposite-face relations for non-adjacent faces according to geometric rules;

4.Finally, output a structured 3D topology representation that includes:

- faces (with explicit names),
- adjacency (specifying shared edges),
- opposite-face pairs,
- vertex_triples (triplets of faces that meet at a single vertex).

[2D Net]
(Insert image1 here)

[Retrieved Geometric Rules]
<related_rules>…</related_rules>

[Retrieved Topologically Isomorphic Sample]
<template image: image2>…</template>

Output Requirements

•Opposite-face pairs must strictly follow the provided geometric rules;

•Each vertex triple must consist of three faces that are pairwise non-opposite and meet at a common vertex;

•All faces must be explicitly named to ensure interpretability.

Please follow the information and format provided in the template (including the description and 3D_structure), and analyze the folded 3D structure of image1 by first determining adjacent faces and then analyzing non-adjacent (opposite) faces. Note that each face in image1 must be referred to using a concrete, explicit name, and the final 3D structure must be output strictly following the template's format.

Figure 10: Prompt for inferring 3D topological relations from the 2D diagram.

You are an expert in geometric folding and texture consistency analysis.

Given:

[Question Q]
(Insert question here)
[2D Net I_2D]
(Insert image of 2D unfolding net)
[Candidate Images O = {O_1, …, O_k}]
(Insert candidate images)
[3D Topology T_3D]
(Insert structured T_3D)

Your tasks are:

1. For each face in each candidate image, identify and describe the visible textures (e.g., animals, icons, symbols, stripe patterns).
2. Using the 2D net I_2D as reference, determine whether each detected texture exists in the corresponding face region of I_2D.
3. A candidate is marked as invalid if any face contains a texture that does NOT appear in the 2D net.
4. For each invalid candidate, explain which face and which texture caused the rejection, including a brief note of the supporting visual evidence (e.g., "distinct red star visible on Face Right, not present in I_2D").
5. The reasoning process must be entirely based on visual evidence from the provided 2D net and candidate images.

[Output Format]

{
  "invalid_candidates": ["O_i", ...],
  "explanations": {
    "O_i": [
      "Face <name> contains texture <description> absent in I_2D (visual evidence: <brief cue>)",
      ...
    ],
    ...
  }
}

[Additional Constraint]

If no invalid candidates are found, output:

{
  "invalid_candidates": [],
  "explanations": {}
}

[Output Requirements]

- Use explicit face names consistent with those in T_3D.
- The explanation for each invalid candidate must reference **specific visual evidence** (color, shape, symbol, or position).
- Do not make assumptions about textures not clearly visible.
- The reasoning must strictly rely on the 2D net and candidate images.
- The output must be valid JSON and include only the keys "invalid_candidates" and "explanations".

Figure 11: Prompt for detecting unseen textures inconsistent with the 2D diagram.

You are an expert in polyhedral folding topology. Your task is not to re-derive or infer the 3D topology from the 2D net.
Instead, you must treat the given 3D topology T_3D as a fixed and authoritative set of constraints. You are allowed to provide explicit reasoning, but your reasoning must remain strictly at a macroscopic level of spatial relations between faces.

[Given]

[2D net I_2D]
(Insert I_2D)
[Remaining Candidates O']
(Insert remaining candidate images)
[3D Topology T_3D obtained from the 2D net]
(Insert T_3D)

[Your Tasks]

For each candidate in O', perform a topological verification using T_3D.
The verification must include:
1. Adjacency Consistency Check
   - Any face pair that is adjacent according to T_3D must appear adjacent in the candidate.
   - Any face pair that appears adjacent in the candidate must not violate an opposite-face constraint.
2. Opposite-Face Consistency Check
   - Faces marked as opposite in T_3D must not appear adjacent in the candidate.
   - Faces marked as opposite must not appear to touch or meet at any visible corner.

[Reasoning Format Requirement]

For each candidate O_j, your reasoning must follow exactly this structure:
Step 1: Observed Adjacencies
- List only the face pairs that are visibly adjacent in the candidate image.
Step 2: Constraint Comparison
- For each observed adjacency, state whether it is allowed or forbidden according to T_3D.
- Explicitly reference the violated constraint if any.
Step 3: Verdict
- State whether the candidate is valid or invalid.
Do not include any reasoning beyond these three steps.

[Output Format]

The final output MUST be valid JSON and include only the following keys:

{
  "invalid_candidates_topo": ["O_j", ...],
  "explanations": {
    "O_j": [
      "Explanation referencing the violated adjacency or opposite-face constraint and the visible evidence in the image."
    ]
  }
}

If no invalid candidates are found, output:

{
  "invalid_candidates_topo": [],
  "explanations": {}
}

[Strict Output Rules]

- Face names must exactly match those defined in T_3D.
- All explanations must be grounded in visible face placement.
- Do not assume unseen connections.
- Do not mention edges, vertices, folding steps, or 2D net reasoning.

Figure 12: Prompt for verifying macro-level spatial topological consistency among surfaces.

You are a visual analysis module specialized in extracting fine-grained visual cues from candidate 3D object images. Your goal is to reduce reliance on implicit prior knowledge and to provide explicit, reliable visual evidence for later spatial reasoning stages. At this stage, you must not perform any topological verification, folding reasoning, or validity judgment. You are only responsible for visual clues extraction and comparison.

[Scope of Analysis]

For each remaining candidate image, perform a fine-grained visual clues analysis focusing on the textures visible on each face.

Your analysis must include:

1. Texture Identification
- Identify the texture content on each visible face (e.g., facial expression, character, icon).
- Describe the texture in concrete visual terms, not symbolic labels unless clearly visible.

2. Texture Type Characterization
- Classify the texture type (e.g., face-like graphic, animal illustration, abstract symbol).
- Note distinctive visual attributes (e.g., orientation, asymmetry, directional features).

3. Inter-Face Texture Relations
- Describe how the texture on one face visually relates to textures on adjacent faces.
- Focus on relative placement, such as:
  - Whether a texture appears near a shared boundary between two faces.
  - Whether a texture element visually approaches or aligns with a corner where multiple faces meet.
  - Do not infer or assume hidden faces or unseen connections.

[Cross-Candidate Comparison]

After analyzing each candidate individually, perform a comparative analysis across candidates:

4. Common Visual Features
- dentify texture patterns or spatial arrangements that are consistent across multiple candidates.

5. Discriminative Visual Differences
- Identify texture placements or relative configurations that differ across candidates.
- Emphasize differences that could later support spatial consistency or inconsistency checks.

These differences should be grounded in visible texture distribution and relative positional cues only.

[Output Format]

- Any reference to other faces must use explicit face names.

Produce a structured, descriptive output with the following format:

```
{
 "per_candidate_visual_clues": {
  "O_j": {
   "visible_faces": [
     {
      "face_texture": "...",
      "texture_type": "...",
      "notable_visual_features": "...",
      "relative_position_cues": "..."   }]}},
 "cross_candidate_analysis": {
  "common_features": [
   "..."
  ],
  "discriminative_differences": [
   {
    "candidates": ["O_j", "O_k"],
    "difference_description": "..."}]}}
[3D Topology T₃D obtained from the 2D net]
"faces": […],
```

[Strict Output Rules]

- All descriptions must be based solely on visible evidence in the images.
- Face names must exactly match those defined in T_3D.
- Use precise, concrete visual language.
- Use specific words to describe each feature, and avoid using terms like "top" or "bottom".

Figure 13: Prompt for visual clues analysis.

You are a verification question generation agent specialized in micro-texture spatial consistency verification for polyhedral folding problems. Your task is to generate verification checkpoints only. You do not solve checkpoints, evaluate candidate correctness, infer answers, or perform reasoning about which option is correct. Your output must consist exclusively of verification instructions.

[Given]

[Remaining Candidate Set O_t]

(Insert remaining candidate set)

[Structured Visual Clues Yvis]

(Insert structured visual cues)

[Verification Question Generation Scope]

- Generate questions based only on explicitly visible evidence provided in Yvis , including exact face names, texture or pattern identities, texture types, relative positional cues between faces, and part-level visual anchors (e.g., eyes, mouth, tongue, gaze direction, curvature, or density regions).

[Spatial Relation Focus]

- Each question must test exactly one fine-grained spatial relation involving textures or their semantic parts, including orientation, proximity, directionality, alignment, curvature, distribution, or relative position.
- The questions must target discrepancies that cannot be captured by coarse face adjacency alone.

[Template-Based Instantiation]

- Generate questions by selecting templates from a predefined question template set.
- Fully instantiate every template slot using exact face names, pattern names, and texture units as defined in Yvis .
- Avoid redundant questions that test the same spatial relation multiple times.

[Restrictions]

- You must generate only verification questions.
- You must not answer any question, judge correctness, infer which candidate is correct, or collapse multiple relations into a single checkpoint.
- Each question must be atomic and focus on a single spatial constraint.

[Output Format]

The final output must be valid JSON and include only a list of generated verification questions.

Each question must specify:

"template_type": the name of the question template used

"verification_question": the fully instantiated verification question

Example format (illustrative only):

```
[{
   "template_type": "Directional-Part Consistency",
    "verification_question": "Is the gaze direction of the eye pattern on Face_Front oriented toward the upper edge relative to the texture on Face_Top after folding?"}]
```

Figure 14: Prompt for verification question generation.

You are a multimodal reasoning agent responsible for detecting micro-texture–level spatial inconsistencies between a given 2D net and multiple 3D candidate options.

[Visual Inputs]

(insert the 2D net, remaining candidate images)

[3D Topology T_3D Derived from the 2D Net]

(Insert T_3D)

[Structured Visual Clues Yvis]

(Insert structured visual cues)

[Micro-Texture Spatial Consistency Verification Questions]

(Insert checklist)

1. Task Objective

• Independent Review: Independently examine each candidate option image in conjunction with the provided visual clues.

• Consistency Determination: Compare the candidate against the 2D Net to determine whether it violates any of the provided Micro-texture Spatial Consistency Verification Questions.

2. Core Verification Principles

• Visual Primacy: Rigorously compare the spatial relationships presented in the 2D Net with those in the candidate images. You must carefully inspect every image and make judgments based on the visual evidence presented across all images, combined with the visual clues Yvis.

• Micro-level Focus: Focus exclusively on the micro-level spatial relationships at the texture layer, including but not limited to:

  • Orientation, Distance, and Directionality.
  • Alignment, Curvature, and Distribution characteristics.
  • The anchoring relationships of texture semantic parts relative to shared edges, vertices, or adjacent faces.

• Strict Boundaries: Do not check any spatial relationships that are not explicitly mentioned in the checklist.

3. Functional Role of T_3D

• Reference for Adjacency Only: The sole role of T_3D is to serve as a reference for valid adjacency relations during reasoning, such as determining which faces are adjacent or opposite after folding, and whether two specific faces share a common edge.

• No Texture Details: T_3D itself does not provide any specific micro-texture spatial relationships, nor does it specify texture orientations, part correspondences, or the positions of semantic components.

• No Over-inference: After obtaining the necessary adjacency information from T_3D, all reasoning must be based entirely on the texture spatial relationships presented in the images. You must not infer any geometric or texture details that are not explicitly given.

4. Evaluation & Elimination Process

• Independent Evaluation: Every verification question must be run independently for each candidate option.

• Elimination Criteria (Invalidation): If any verification question reveals a spatial inconsistency—where the micro-texture relationship observed in the candidate contradicts the relationship required by the constraints of the 2D Net (combined with the adjacency permitted by T_3D)—the candidate must be marked as Invalid and eliminated.

Figure 15: Prompt for verifying micro-level spatial consistency.

You are a reasoning agent tasked with integrating multi-stage reasoning evidence into a single, coherent decision rationale.

Input: An integrated context $\mathcal{C}$_integ = {$\boldsymbol{Y}$_sem, $\boldsymbol{Y}$_macro, $\boldsymbol{Y}$_micro}, which contains the semantic-level, macro-structural, and micro-textural reasoning outputs from previous stages.

Task: Aggregate the provided evidence to produce a final reasoning chain $\boldsymbol{Y}$_final that explains why the selected option is correct and why other options are invalid, if applicable.

Constraints:
• Use only information in $\mathcal{C}$_integ; do not introduce new observations or interpretations.
• Follow a coarse-to-fine fusion order: semantic → macro → micro.
• Options eliminated at earlier stages must remain invalid.
• Do not re-evaluate or re-derive intermediate results.

Output:
Produce a concise, step-by-step explanation that integrates all stages and concludes with the final decision or explicitly states ambiguity if it remains.

Figure 16: Prompt for integrating multi-stage reasoning outputs.