WebSpatial: A Benchmarking Framework of Spatial Intelligence via Web-based Runtime Environments

Shaofei Liu[1,*] **Weichen Bi**[1,*] **Yun Ma**[1,†]

[1]Institute for Artificial Intelligence, Peking University
Beijing, China

liushaofei26@stu.pku.edu.cn, biweichen@pku.edu.cn, mayun@pku.edu.cn

[*]Equal contribution. [†]Corresponding author.

Abstract

Spatial intelligence is a fundamental capability of embodied Artificial Intelligence systems, and its reliable assessment and optimization require well-designed benchmarks. Existing benchmarks for spatial intelligence largely rely on video datasets or procedurally generated data, which limits data scalability and ecological validity. To address these limitations, we propose to automatically extract spatial information directly from executable environments. Specifically, we design WebSpatial, a benchmarking framework built upon web runtime environments, which enables automatic acquisition of spatial data by parsing underlying code and interaction processes. It performs runtime scene graph parsing on web graphics applications to automatically extract visual attributes, world coordinates, and user interaction event sequences of objects, while constructing a dedicated spatial computation tool library. The modular generation pipeline integrates LLM-based task parsing and planning, and tool-assisted execution to automatically generate verifiable question-answer pairs. We conduct experiments in method effectiveness validation and benchmark evaluation. For validation, we demonstrate that WebSpatial can reliably construct diverse and scalable spatial tasks grounded in both static and dynamic scenes, covering spatial perception including color, shape and quantity, and spatial reasoning including relation assessment, direction transformation and mental rotation. We develop a small-scale benchmark named WebSpatialBench based on WebSpatial, and use it to evaluate 10 multimodal large language models, revealing their strengths and limitations in spatial problem-solving.[1]

1 Introduction

Spatial intelligence refers to the ability of AI systems to perceive, understand, reason, and interact in 3D environments, where performance depends on runtime context, interaction sequences, and system dynamics. Although recent multimodal large language models (MLLMs) have achieved strong results in language and 2D visual understanding[1, 2], their capabilities in 3D spatial reasoning remain limited[3, 4]. For evaluation, existing benchmarks are largely built on 2D images and videos[5, 6, 7] or pre-collected 3D data with heavy reliance on manual annotation, resulting in limited scalability and diversity. Procedurally generated datasets[8, 9, 10], in contrast, enable larger-scale and greater diversity, but often lack ecological validity and fail to capture the complexity of real-world spatial interactions. Such paradigms are difficult to extend and evolve over time, highlighting the lack of an automated and scalable approach for constructing spatial intelligence benchmarks. Such limitations

[1]https://github.com/ShaofeiLiu-sofia/WebSpatial/

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

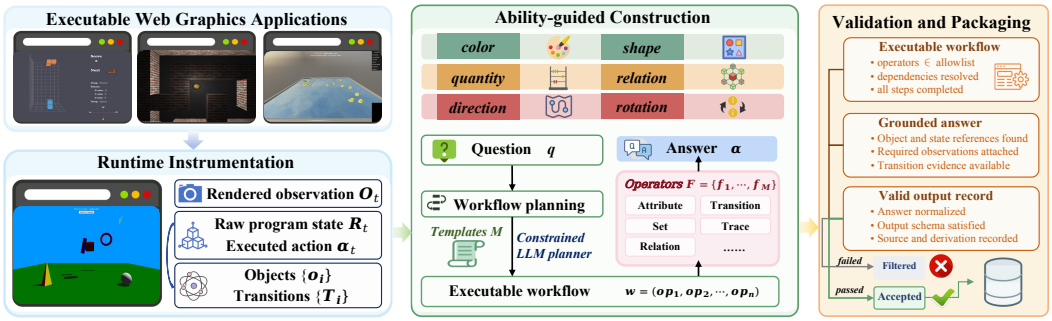

Figure 1: Overview of WebSpatial. WebSpatial instruments executable web graphics applications, aligns rendered observations with runtime states and interaction transitions, normalizes the evidence, and constructs validated spatial-intelligence instances with programmatic answer oracles and provenance.

hinder the continuous evolution of evaluation frameworks, highlighting the need for an automated, scalable and ecologically grounded approach to spatial intelligence benchmarking.

To address these issues, this paper proposes a scalable automated benchmark construction method, using web graphics applications as the specific implementation carrier to obtain spatial data from the operating environment. In web graphics applications, spatial information are typically defined through explicit variables, allowing them to be programmatically perceived, recorded, and parsed[11, 12]. User interactions reflect how humans perceive, interpret, and reason about spatial environments, capturing essential aspects of human spatial understanding and mental modeling. Therefore, conducting spatial intelligence benchmark research based on web applications can efficiently extract scene information including object positions, shapes, colors and users' interactive behavior data, which can more realistically reflect the model's performance in actual spatial environments. The open nature of web technologies provides high extensibility, allowing new scenarios to be readily incorporated and evaluation tasks to be updated promptly as research advances.

In this paper, we design WebSpatial, a framework for automatically constructing spatial intelligence benchmark via web-based runtime environments. Figure 1 provides an overview of the Webspatial framework. It adopts a hierarchical taxonomy with 2 categories and 6 subcategories that characterize spatial perception and spatial reasoning capabilities as shown in Table 2. Guided by this taxonomy, WebSpatial selects or composes executable task workflows over the captured runtime evidence, producing questions and programmatic oracle answers that target one or more spatial abilities. At the data collection stage, WebSpatial executes each application in a controlled web-based runtime environment, extracts state screenshots, object states, user actions, and state diffs, and converts them into a unified spatial state and trace representation. It then applies a library of executable spatial operators to generate questions, answers, and oracle evidence. We adopt a multi-module generation architecture, comprising an LLM-based task parser, a rule-based task planner, tool-assisted execution, and result recording. This automated pipeline significantly reduces cost while producing scalable, real-world data that reflects human spatial understanding and reasoning.

Based on this method, we build WebSpatialBench, a spatial intelligence benchmark dataset comprising 1,042 QA instances collected from 30 executable web environments. Each QA pair may involve 1 or multiple spatial dimensions, reflecting the compositional nature of real-world spatial reasoning. Our evaluation include verifying the effectiveness of the proposed method and measuring 10 state-of-the-art MLLMs on WebSpatialBench. Experimental results show that the best-performing model achieves only around 72.77% accuracy, indicating substantial gaps in current spatial reasoning capabilities. While models perform relatively well on basic perception tasks, they struggle significantly with mental modeling tasks like rotation, highlighting a big gap between spatial understanding and imagination.

Contributions of this paper are as follows:

- Based on core capabilities of spatial intelligence, we introduce **WebSpatial**, a unified runtime-instrumented framework that automatically extracts spatial information from web graphics applications and converts it into structured spatial intelligence benchmark content with executable provenance.

- We construct **WebSpatialBench** using WebSpatial, comprising 1,042 QA instances from 30 executable web environments and covering color, shape, quantity, relation, direction, and rotation.
- Comprehensive and systematic evaluation are conducted, with detailed analyses of their strengths, limitations, and common failure modes in spatial tasks.

2 Related Work

In this section, we review prior work from two perspectives: the characterization of spatial capabilities and the construction of benchmarks for evaluation.

2.1 Spatial Intelligence

Spatial intelligence was originally introduced in cognitive psychology as a central component of the Theory of Multiple Intelligences[13], describing humans' ability to perceive, mentally transform, and reason about 3D environments. Current research in spatial intelligence focuses on spatial perception[14, 15], reasoning[16, 17], and action[18, 19]. World Labs[20] reports systems capable of generating navigable, persistent, and user-controllable 3D worlds. Wang et al. propose the Web World Model (WWM)[21], which explicitly encodes world states and rules as executable web code and delegates high-level narration and decision-making to large models, providing a scalable and controllable intermediate world modeling framework.

2.2 Benchmarks for Spatial Intelligence

Table 1: Comparison of representative spatial-intelligence benchmarks.

Benchmark	Data Source	Observation	Label Source	Capabilities
VSI-Bench	Egocentric indoor videos	Video observation	Automatic QA generation with human route annotation and review	Relational reasoning: distance, direction, size. Egocentric–allocentric transformation: layout, perspective-taking
MINDCUBE	Limited multi-view scene images	Multi-view observation	Human annotation with automatic QA generation	Cognitive mapping, perspective-taking, mental simulation
Spatial457	Synthetic 3D scenes	Static image observation	Fully programmatic generation	3D spatial reasoning: multiple objects, 2D locations, 3D locations, 3D orientations
SpatialViz-Bench	Programmatically generated visual puzzles	Controlled visual puzzle instances	Puzzle-generation rules and expert-designed task structure	Mental rotation, folding, penetration, animation
iVISPAR	Simulated fixed 4×4 board	Board-state interaction	Simulator-based automatic generation	Logical planning, spatial awareness, multi-step reasoning
WebSpatialBench	Web-based runtime environments	Scene-state observation	Runtime-extracted automatic generation	Color, shape, quantity, relation, direction, rotation

To systematically compare model capabilities across different dimensions of spatial intelligence, some benchmarks are proposed[5, 22, 6, 23, 10, 24]. One representative line of research constructs benchmarks from video data, typically relying on manually designed question templates and annotated answers. VSI-Bench[5] builds over 5,000 QA pairs from nearly 290 real indoor scene videos, covering configuration, measurement estimation, and spatiotemporal reasoning tasks to assess models' spatial perception, memory, and reasoning under continuous visual input. MINDCUBE[22] organizes multi-view image sets together with 21,154 spatial reasoning questions, aiming to evaluate a model's ability to construct internal spatial mental models under limited observations. Synthetic datasets are commonly used to reduce annotation cost and expand task diversity. Spatial457[10] defines 4 core capabilities and systematically organizes them into 5 difficulty levels and 7 question types, programmatically generating an unbiased and scalable dataset for diagnosing multimodal large models' 6D spatial reasoning abilities.

Unlike prior works, our approach enables automatic and scalable construction of data from executable web environments, reducing manual effort while supporting continuous expansion.

Table 2: Taxonomy of core capabilities evaluated in WebSpatialBench, including category, subcategory and description.

Category	Subcategory	Description
Spatial Perception	Color	Identifying specific objects' color.
	Shape	Recognizing object shapes from visual representations.
	Quantity	counting the number of objects that satisfy specific spatial conditions.
Spatial Reasoning	Relation	Reasoning about spatial relationships among objects and between objects and surrounding scene.
	Direction	Inferring the directional transformation of objects.
	Rotation	Understanding changes in object orientation under rotational transformations.

3 WebSpatial

In this section, we present WebSpatial, an automated framework for benchmarking spatial intelligence via web-based runtime environment.

3.1 Overview

Given a web environment, the goal of WebSpatial is to automatically construct QA pairs that evaluate spatial perception and reasoning capabilities. The key idea is to directly leverage the runtime environment to obtain structured scene information and generate QA pairs, enabling scalable and realistic benchmark construction.

The framework consists of two stages: (1) data acquisition and (2) QA generation. In the first stage, given a web environment E, WebSpatial launches the application in a controlled web runtime and captures a rendered observation O_t together with its aligned raw runtime state R_t. We parse the runtime environment E to extract a structured scene representation $\mathcal{S}$, including object attributes, spatial relations, and interaction-driven state changes. This stage provides a unified foundation for subsequent task construction. In the second stage, QA pairs are generated. A query q is parsed and mapped to an executable workflow $w = (op_1, op_2, \ldots, op_n)$, where each op_i denotes an atomic operation. The answer a is obtained by executing w over $\mathcal{S}$.

From the collected runtime evidence, WebSpatial produces benchmark instances

$$x = \langle O, q, a, w, m \rangle, \tag{1}$$

where O contains one or more rendered observations, q is an ability-guided question, a is the normalized target answer, w is the executable oracle workflow, and m records provenance such as the source application, scene state, action trace, question type, and spatial-ability labels. Figure 1 summarizes the stage-level pipeline.

3.2 Capabilities for Spatial Intelligence

To systematically characterize the performance of models in spatial intelligence, we structurally categorize spatial intelligence abilities based on the theory of multiple intelligences as shown in Table 2. Spatial intelligence is divided into two main dimensions: spatial perception and spatial reasoning, where spatial perception includes color, shape, quantity, and spatial reasoning includes several ability dimensions such as spatial relationship judgment, directional transformation, and mental rotation.

3.3 Data Acquisition

WebSpatial normalizes extracted information into a unified spatial state representation. Each object is represented as

$$o_i = \langle id, type, attr, p, r, s, v \rangle, \tag{2}$$

Algorithm 1 Data Collection

Require: Web application URL u, scene entry p
Ensure: Structured data D

Launch browser and navigate to u
Wait until the web page and 3D scene are fully loaded
$\mathcal{O} \leftarrow \emptyset$, $E \leftarrow \emptyset$
{Scene extraction}
Access scene object S from entry p
for each node o in S **do**
  if o is a geometric object **then**
    Extract attributes of o, local position $\text{pos}(o)$ and global position $\text{wpos}(o)$
    $\mathcal{O} \leftarrow \mathcal{O} \cup \{o\}$
  end if
end for
if hierarchical serialization is available **then**
  Align extracted attributes with scene structure
  $D_{\text{scene}} \leftarrow$ serialized scene representation
else
  $D_{\text{scene}} \leftarrow \mathcal{O}$
end if
{Interaction recording}
while application is active **do**
  if interaction event detected **then**
    Capture current state $\mathcal{S}_t$
    Record action a_t and timestamp t
    Wait for scene update
    Capture next state $\mathcal{S}_{t+1}$
    $E \leftarrow E \cup \{(a_t, \mathcal{S}_t, \mathcal{S}_{t+1})\}$
  end if
end while
{Data integration}
if $E \neq \emptyset$ **then**
  $D \leftarrow \text{Merge}(D_{\text{scene}}, E)$
else
  $D \leftarrow D_{\text{scene}}$
end if
Remove invalid or redundant references in D
Save D in structured format
return D

where id identifies the object, $type$ records geometry or semantic role, $attr$ stores visual attributes, p stores position in the relevant coordinate system, r stores orientation or rotation when available, s stores size or grid occupancy, and v records visibility or view-dependent metadata. Pairwise and higher-order structures, such as adjacency, containment, contact, alignment, and neighborhood relations, are derived from this state representation.

For dynamic sources, WebSpatial records a transition as

$$T_t = \langle S_t, a_t, S_{t+1}, \Delta_t, O_t, O_{t+1} \rangle, \tag{3}$$

where S_t and S_{t+1} are normalized states, a_t is the executed action, Δ_t is the state diff, and O_t, O_{t+1} are visual observations before and after the transition.

This unified representation provides the foundation for both spatial perception and reasoning tasks.

The data collection process is shown in Algorithm 1.

3.4 QA Generation

The QA generation stage has 2 phases, including task parsing and planning, and tool-based execution. It parses a query q to a workflow $w = (op_1, \ldots, op_n)$, and executes w with tools to obtain the answer a, forming the QA pair (q, a).

3.4.1 Task Parsing and Planning

Given a q, WebSpatial constructs an executable reasoning procedure through 2 sequential steps: task parsing and workflow planning.

Task parsing. The input query q is analyzed to identify its semantic components, including target objects, attribute or spatial relation constraints. Based on the spatial-capability taxonomy, each query may target one or more subcategories and is further classified as either a static or a dynamic task.

Static tasks focus on spatial perception and basic reasoning over a single scene state. These tasks include object attribute recognition, such as color and shape identification, object counting under attribute or relational constraints, and spatial relation understanding between objects. Dynamic tasks involve reasoning over state changes induced by interactions. These tasks require modeling transformations such as directional movement or rotation across different time steps, where the answer depends on the evolution of the scene rather than a single snapshot.

Workflow planning. After parsing, the query is decomposed into a sequence of atomic operations with explicitly defined inputs and outputs. Each operation corresponds to a basic spatial computation step. The resulting sequence $w = (op_1, op_2, \ldots, op_n)$ represents the reasoning procedure.

To improve efficiency and scalability, WebSpatial maintains a library of workflow templates that capture common query patterns. Given a query, we first attempt to match it to an existing template and instantiate the corresponding workflow through parameter binding. For queries that cannot be directly matched, LLM generates workflow by decomposing the query into similar substructures guided by existing templates. Successfully constructed workflows are incorporated into the template library, enabling iterative refinement and expansion over time. This mechanism allows the framework to balance efficiency with flexibility, supporting both standardized and previously unseen spatial queries.

3.4.2 Tool-based Execution

In the execution stage, the workflow generated during task planning is carried out as a sequence of spatial operations. To support diverse spatial tasks in a unified manner, we define a set of executable operators that act on scene objects and interaction signals. These operators provide the basic primitives for spatial reasoning and can be composed to handle both simple and multi-step queries. The operators can be grouped into 5 categories according to their functionality.

Attribute extraction. The attribute extraction operators map object instances to their semantic properties, such as color or shape. Given an object set $\mathcal{O}$, the operator performs a projection from object identifiers to attribute values:

$$\text{attr}(o_i) \rightarrow v_i$$

where v_i denotes the queried attribute. This step standardizes heterogeneous visual properties into comparable symbolic representation, enabling downstream reasoning operations such as filtering and aggregation.

Object filtering. Filtering operators select subsets of objects that satisfy a set of logical constraints. Formally, given a predicate function f, filtering is defined as:

$$\mathcal{O}' = \{o \in \mathcal{O} \mid f(o) = \text{true}\}$$

where f may encode attribute conditions (e.g., color, shape) or spatial constraints.

Conditions may involve attribute constraints or spatial predicates. By iteratively applying filtering operations, the system progressively narrows down the candidate set, enabling precise localization of target objects.

Counting. The counting operator compute statistical properties over a set of objects. The most common operation is cardinality estimation:

$$\text{count}(\mathcal{O}') = |\mathcal{O}'|$$

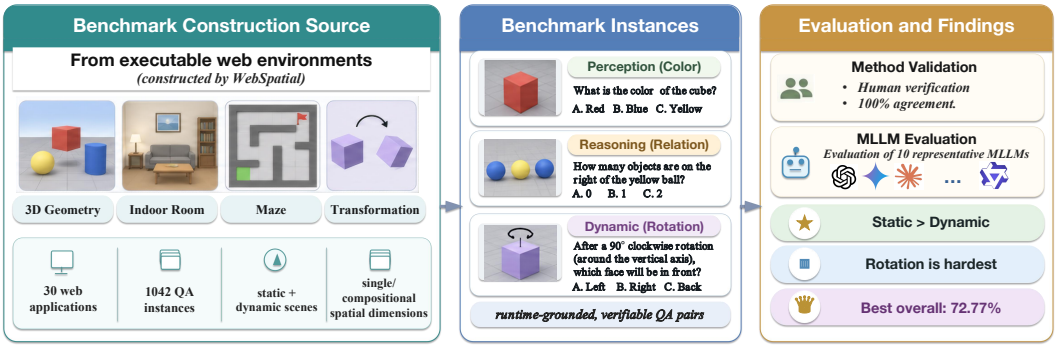

Figure 2: Overview of WebSpatialBench.

It's commonly used in tasks that require numerical reasoning over filtered results.

Spatial relation computation. The relation operator evaluate the relative spatial configuration between objects based on their geometric properties. Given 2 objects o_i and o_j, their spatial relation is determined by comparing their positions and spatial extents along different axes. Thus, it allows evaluation of directional relations of left, right, top, below, above, and behind.

Transformation and interaction Analysis. For dynamic tasks, the execution process incorporates this operator that analyze state changes induced by interaction. Given a sequence of states $\{\mathcal{S}_t\}$, the operator infer transformations by analyzing temporal differences:

$$\Delta(o) = \mathcal{S}_{t+1}(o) - \mathcal{S}_t(o)$$

By operating over state transitions, this operator enables reasoning about temporal evolution in interactive environments.

4 Experiments

In this section, we first assess WebSpatial's effectiveness as an automated benchmark-construction method, and then use the validated benchmark as a diagnostic instance to evaluate the spatial capabilities of current MLLMs.

4.1 Evaluation Protocol

To evaluate whether WebSpatial can reliably transform executable web-based runtime environments into spatial QA tasks, we instantiate the construction pipeline on a collection of 30 web graphics applications.

Environment Selection. To construct spatial intelligence tasks from realistic environments, we select web graphics applications that contain visually identifiable objects, explicit spatial structures, or observable state changes. Our benchmark includes both static and dynamic applications, with many of the selected 3D applications implemented[25] using Three.js[26]. It provides an official examples library that contains a large number of 3D scenes, which is well suited to serve as the experimental environment for the construction of spatial intelligence tasks. For the selection of sample applications, this study establishes specific screening criteria to ensure that the selected scenes can support the construction of spatial intelligence tasks. The main criteria are specified as follows:

1) Each scene contains at least 2 objects that can be clearly recognized by human vision.

2) There exist explicit spatial relationships or attribute differences among objects, such as position, color distinctions or other characteristics.

3) Scenes for dynamic tasks are required to support either viewpoint changes or user interactions.

WebSpatialBench. For each environment, WebSpatial programmatically access and monitor web graphics applications using Algorithm 1 to obtain structured scene data. It extracts runtime scene

or state information, instantiates capability-oriented task templates, executes the corresponding workflows to obtain grounded answers, and retains QA pairs that pass the validity and visual-answerability checks.

The resulting WebSpatialBench shown in Figure 2 contains 1,042 QA pairs constructed from 30 environments and covers 6 spatial capabilities. Each instance may involve one or multiple evaluation dimensions. It's used below both to validate the construction process and to further diagnose the spatial capabilities of current MLLMs.

4.2 Method Validation

We validate the construction method by examining whether the generated data are executable, grounded in runtime states, visually answerable from the provided observations, and associated with correct reference answers.

Evaluation Setup. For each generated sample, we first execute the planned workflow and reject samples whose operations cannot be completed or whose outputs cannot be parsed. For successfully executed workflows, we compare the generated record with the runtime state and retain only samples that satisfy the predefined validity and visual-answerability criteria. To complement the automated checks, we manually inspect all QA pairs across the six capability categories and both static and dynamic settings. Each sample is examined for question clarity, consistency between the visual input and question, and correctness of the reference answer. Each sample is independently evaluated by 2 annotators with relevant background knowledge, who assess the correctness and consistency of the question formulation, task logic, and corresponding answer. To reduce subjective bias, annotations are conducted independently, and disagreements are resolved through discussion to reach consensus.

Results Analysis. 134 candidate items were removed because the rendered evidence was blurry, occluded, viewpoint-ambiguous, or judged by both annotators to be too trivial, overly complex, or not diagnostically meaningful. The remaining 1,042 QA instances constitute the final dataset. We then compare the human-validated results with the automatically generated answers on a per-sample basis. The results show complete agreement across all validation samples, yielding an accuracy of 100%. This indicates that the proposed method achieves high accuracy and stability within the evaluated scope, and is effective for constructing spatial intelligence benchmark data.

4.3 Evaluation of MLLMs' Spatial Capabilities

Having validated the construction pipeline and the resulting QA pairs, we further use WebSpatial-Bench to diagnose the spatial capabilities of current MLLMs.

Evaluation Setup. We evaluate 10 representative MLLMs[27, 28, 29, 30, 31, 32] on the same 1,042 multiple-choice QA instances. For static tasks, each model receives a question, four answer options, and a single rendered screenshot. For dynamic tasks, direction questions are accompanied by temporally ordered multi-screenshot trajectory segments, while rotation questions use paired screenshots representing the states at T_0 and T_1. The recorded interaction events and runtime metadata are used only to construct verifiable reference answers and are never provided to the evaluated models.

All models are evaluated using the same prompt template, QA instances, option ordering, and response-parsing procedure. We set the temperature to 0 and instruct each model to return one or more option labels from A, B, C, and D. The evaluation script maps the predicted label to the corresponding option and compares it with the runtime-derived reference answer. We report the median result over three runs for each model, and present instance-level accuracy for each capability. Static and Dynamic Avg. denote the accuracies aggregated over all QA instances in the corresponding groups.

Results Analysis. Table 3 reports the result of the evaluation. According to the static and dynamic properties of scenes, the entire dataset is categorized into static and dynamic groups for evaluation purposes. For each metric, the best-performing model is highlighted in bold, and the second-best is underlined; ties receive the same formatting.

1) How well do MLLMs perform on spatial intelligence tasks? We observe that the overall spatial intelligence of current MLLMs remains limited. The best-performing model achieves an average

Table 3: Fine-grained performance (%) on WebSpatialBench. Best and second-best model results in each column are shown in bold and underlined, respectively.

	Static Group					Dynamic Group		
Model	**Color**	**Shape**	**Count**	**Relation**	**Avg.**	**Direction**	**Rotation**	**Avg.**
Baseline (Random)	24.39	25.00	26.32	23.91	25.00	24.97	25.93	25.06
Seed-2.0-Mini	**68.29**	**100.00**	<u>77.19</u>	76.09	<u>75.66</u>	73.30	35.80	69.89
Gemini 2.5 Pro	<u>63.41</u>	<u>87.50</u>	68.42	67.39	67.76	65.39	55.56	64.49
Claude Sonnet 4.6	58.54	**100.00**	45.61	54.35	54.61	<u>75.77</u>	<u>58.02</u>	<u>74.16</u>
GPT-5.4	<u>63.41</u>	<u>87.50</u>	**80.70**	**95.65**	**80.92**	48.58	11.11	45.17
GPT-5	51.22	**100.00**	36.84	<u>82.61</u>	57.89	56.24	28.40	53.71
GPT-5-Mini	56.10	<u>87.50</u>	52.63	<u>82.61</u>	64.47	45.24	11.11	42.13
Qwen3 VL 8B Thinking	60.98	<u>87.50</u>	70.18	65.22	67.11	46.97	**60.49**	48.20
Qwen3.6 Plus	56.10	<u>87.50</u>	59.65	54.35	58.55	62.42	18.52	58.43
Qwen3 VL 32B Instruct	48.78	75.00	59.65	50.00	54.61	**76.14**	**60.49**	**74.72**
Qwen3 VL 8B Instruct	36.59	75.00	36.84	45.65	41.45	39.93	<u>58.02</u>	41.57

accuracy of only 72.77%, indicating that reliable spatial understanding in realistic interactive settings has not yet been fully achieved.

Model performance varies substantially across different MLLMs on WebSpatialBench. Seed-2.0-Mini achieves the highest overall performance (72.77%), followed by Gemini 2.5 Pro (66.13%). GPT-5.4 achieves the best performance on static tasks, while Qwen3 VL 32B Instruct performs best on dynamic tasks. In general, models that have undergone vision-language alignment and task-specific optimization tend to outperform more general-purpose multimodal models.

For the GPT series, the models demonstrate relatively strong average performance, but exhibit notable discrepancies between static and dynamic tasks. In particular, GPT-5.4 achieves the best performance on static tasks (80.92%), significantly outperforming GPT-5 (57.89%). However, its performance degrades considerably on dynamic tasks, suggesting limitations in modeling temporal spatial changes.

The results for the Qwen VL series further reveal the impact of model scale and reasoning mechanisms on spatial reasoning performance. Larger models (32B-Instruct) consistently outperform smaller counterparts (8B-Instruct) on overall spatial tasks. Models equipped with explicit reasoning mechanisms (8B-Thinking) demonstrate stronger performance than their non-reasoning variants (8B-Instruct).

2) How do MLLMs perform across different spatial capabilities and task types? As shown in Table 3, static spatial perception is largely well handled by current MLLMs, but performance degrades as task complexity increases. GPT-series models demonstrate strong performance on static tasks, particularly in spatial relation reasoning, but degrade significantly on dynamic tasks. In contrast, Qwen-VL models exhibit relatively stronger performance on dynamic tasks, especially in transformation-related reasoning.

In particular, shape recognition is nearly saturated, while color recognition exhibits significantly higher variance, indicating sensitivity to visual noise and rendering variations in web environments. Counting and spatial relation tasks reveal partial limitations over complex layouts, lacks robustness in multi-object or ambiguous-view scenarios.

A more pronounced gap emerges between static and dynamic spatial understanding. While several models achieve strong performance on direction transformation tasks (over 70%), models that excel in static perception do not consistently transfer this advantage to dynamic settings. Rotation transformation remains the primary bottleneck. Most models perform poorly, with accuracy dropping to below 30% in some cases, and even the best models reaching only 60.49%. This highlights a fundamental limitation in spatial abstraction and mental transformation capabilities.

Overall, current MLLMs demonstrate strong performance in basic perception and simple spatial reasoning, but struggle with compositional reasoning and dynamic spatial transformations, especially under rotation.

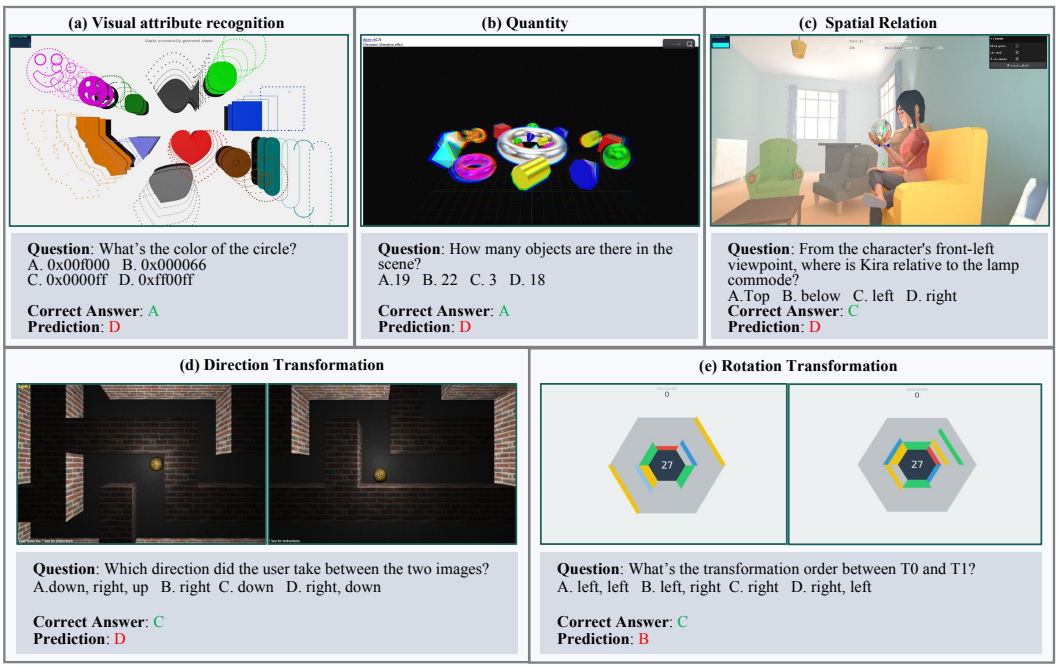

Figure 3: Representative failure cases across 5 specific tasks, illustrating common errors in visual attribute recognition, quantity, spatial relation, direction and rotation transformation.

3) What types of errors do MLLMs make in WebSpatialBench? In order to deeply analyze the capability boundaries and potential shortcomings of MLLMs in spatial intelligence tasks, we present representative error cases in Figure 3 to illustrate common error patterns of MLLMs across different tasks.

Visual Attribute Recognition. Although color recognition is generally simple, performance degrades in cluttered scenes with overlapping objects and visually similar distractors, leading to confusion between target and non-target objects.

Quantity. Models struggle with accurate counting when object boundaries are ambiguous or partially occluded, resulting in undercounting or overcounting.

Spatial Relation. Errors are mainly caused by viewpoint ambiguity and occlusion, making it difficult for models to correctly align spatial relationships under perspective changes.

Direction Transformation. Models capture overall movement trends but fail to precisely infer exact action sequences, especially for subtle or boundary-level changes.

Rotation Transformation. Rotation tasks require tracking object states and reasoning over multiple transformations, posing significant challenges for current MLLMs.

5 Conclusion

We introduce WebSpatial, an automated framework for constructing spatial intelligence benchmarks from web-based runtime environments, and build WebSpatialBench with 1,042 QA instances. Extensive evaluation shows that current MLLMs achieve reasonable performance in basic perception but exhibit clear limitations in spatial reasoning, particularly in dynamic and rotation-related tasks. These findings highlight a gap between spatial perception and higher-level spatial reasoning.

Limitations and Future Work A limitation is that our dataset is constructed from a limited number of web applications, and many scenes remain simple. WebSpatial supports incremental expansion to new executable web graphics applications. Broader generality can be further examined as the framework is extended to more diverse scenes, interaction patterns, and task difficulties. The pipeline has already been applied to more complex applications, including 2048, Sokoban, and Tetris-style

environments. Future work will continue to expand the scale and complexity of the environment collection.

6 Acknowledgments

This work was supported by New Generation Artificial Intelligence-National Science and Technology Major Project (No. 2025ZD0123800), the National Natural Science Foundation of China (No. 62595734), the Key Laboratory of High Confidence Software Technologies (Peking University), the Ministry of Education, and the Center for Data Space Technology and System, Peking University.

References

[1] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: towards general-purpose vision-language models with instruction tuning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

[2] Xiang Yue, Yuansheng Ni, Tianyu Zheng, Kai Zhang, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9556–9567, 2024. doi: 10.1109/CVPR52733.2024.00913.

[3] Haider Al-Tahan, Quentin Garrido, Randall Balestriero, Diane Bouchacourt, Caner Hazirbas, and Mark Ibrahim. Unibench: Visual reasoning requires rethinking vision-language beyond scaling. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 82411–82437. Curran Associates, Inc., 2024. doi: 10.52202/079017-2620. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/96271227d3e204501d199433e56af289-Paper-Datasets_and_Benchmarks_Track.pdf.

[4] 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 A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 75392–75421. Curran Associates, Inc., 2024. doi: 10.52202/079017-2400. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/89cc5e613d34f90de90c21e996e60b30-Paper-Conference.pdf.

[5] Jihan Yang, Shusheng Yang, Anjali W Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. 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, 2025.

[6] Wenqi Wang, Reuben Tan, Pengyue Zhu, Jianwei Yang, Zhengyuan Yang, Lijuan Wang, Andrey Kolobov, Jianfeng Gao, and Boqing Gong. Site: towards spatial intelligence thorough evaluation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9058–9069, 2025.

[7] Yuyou Zhang, Radu Corcodel, Chiori Hori, Anoop Cherian, and DING ZHAO. Spin-bench: Perspective and rotation as a lens on spatial reasoning in vlms. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust, editors, *International Conference on Learning Representations*, volume 2026, pages 70072–70141, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/724be4472168f31ba1c9ac630f15dec8-Paper-Conference.pdf.

[8] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2901–2910, 2017. doi: 10.1109/CVPR.2017.215.

[9] Jiahui Zhang, Yurui Chen, Yueming Xu, Ze Huang, Jilin Mei, Chunhui Chen, Yanpeng Zhou, Yu-Jie Yuan, Xinyue Cai, Guowei Huang, Xingyue Quan, Hang Xu, and Li Zhang. From flatland to space: Teaching vision-language models to perceive and reason in 3d. In *Advances in Neural Information Processing Systems*, 2025. doi: 10.52202/085713-5180.

[10] Xingrui Wang, Wufei Ma, Tiezheng Zhang, Celso M de Melo, Jieneng Chen, and Alan Yuille. Spatial457: A diagnostic benchmark for 6d spatial reasoning of large mutimodal models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24669–24679, 2025.

[11] Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 52040–52094. Curran Associates, Inc., 2024. doi: 10.52202/079017-1650. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/5d413e48f84dc61244b6be550f1cd8f5-Paper-Datasets_and_Benchmarks_Track.pdf.

[12] Div Garg, Diego Caples, Andis Draguns, Nikil Ravi, Pranav Putta, Naman Garg, Prannay Hebbar, Youngchul Joo, Jindong Gu, Charles London, Christian Schroeder de Witt, and Sumeet Motwani. Real: Benchmarking autonomous agents on deterministic simulations of real websites. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen, editors, *Advances in Neural Information Processing Systems*, volume 38, Main Conference. Curran Associates, Inc., 2025. doi: 10.52202/085713-4536. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/c63819755591ea972f8570beffca6b1b-Paper-Datasets_and_Benchmarks_Track.pdf.

[13] Howard Gardner. The theory of multiple intelligences. *Annals of dyslexia*, pages 19–35, 1987.

[14] Justin Lazarow, David Griffiths, Gefen Kohavi, Francisco Crespo, and Afshin Dehghan. Cubify anything: Scaling indoor 3d object detection. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22225–22233, 2025.

[15] Kaixin Yao, Longwen Zhang, Xinhao Yan, Yan Zeng, Qixuan Zhang, Lan Xu, Wei Yang, Jiayuan Gu, and Jingyi Yu. Cast: Component-aligned 3d scene reconstruction an rgb image. *ACM Transactions on Graphics (TOG)*, 44(4):1–19, 2025.

[16] Diankun Wu, Fangfu Liu, Yi-Hsin Hung, and Yueqi Duan. Spatial-mllm: Boosting mllm capabilities in visual-based spatial intelligence. *arXiv preprint arXiv:2505.23747*, 2025.

[17] Yongsen Mao, Junhao Zhong, Chuan Fang, Jia Zheng, Rui Tang, Hao Zhu, Ping Tan, and Zihan Zhou. Spatiallm: Training large language models for structured indoor modeling. In *Advances in Neural Information Processing Systems*, 2025.

[18] Jingkang Yang, Shuai Liu, Hongming Guo, Yuhao Dong, Xiamengwei Zhang, Sicheng Zhang, Pengyun Wang, Zitang Zhou, Binzhu Xie, Ziyue Wang, et al. Egolife: Towards egocentric life assistant. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 28885–28900, 2025.

[19] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[20] World Labs Team. Marble: A Multimodal World Model. https://www.worldlabs.ai/blog/marble-world-model, November 2025. Accessed: 2026-5-02.

[21] Jichen Feng, Yifan Zhang, Chenggong Zhang, Yifu Lu, Shilong Liu, and Mengdi Wang. Web world models. *arXiv preprint arXiv:2512.23676*, 2025.

[22] Baiqiao Yin, Qineng Wang, Pingyue Zhang, Jianshu Zhang, Kangrui Wang, Zihan Wang, Jieyu Zhang, Keshigeyan Chandrasegaran, Han Liu, Ranjay Krishna, et al. Spatial mental modeling from limited views. In *Structural Priors for Vision Workshop at ICCV'25*, 2025.

[23] Weichen Zhang, Zile Zhou, Xin Zeng, Liu Xuchen, Jianjie Fang, Chen Gao, Jinqiang Cui, Yong Li, Xinlei Chen, and Xiao-Ping Zhang. Open3d-vqa: A benchmark for embodied spatial concept reasoning with multimodal large language model in open space. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 12784–12791, 2025.

[24] Julius Mayer, Mohamad Ballout, Serwan Jassim, Farbod Nosrat Nezami, and Elia Bruni. ivispar—an interactive visual-spatial reasoning benchmark for vlms. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 26745–26769, 2025.

[25] Weichen Bi, Yun Ma, Deyu Tian, Qi Yang, Mingtao Zhang, and Xiang Jing. Demystifying mobile extended reality in web browsers: How far can we go? In *Proceedings of the ACM Web Conference 2023*, WWW '23, page 2960–2969, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450394161. doi: 10.1145/3543507.3583329. URL `https://doi.org/10.1145/3543507.3583329`.

[26] Brian Danchilla. Three. js framework. In *Beginning WebGL for HTML5*, pages 173–203. Springer, 2012.

[27] ByteDance Seed Team. Seed-2.0-mini. `https://seed.bytedance.com/en/seed2`, 2026.

[28] Google DeepMind. Gemini 2.5 pro. `https://ai.google.dev/gemini-api/docs/models/gemini-2.5-pro`, 2025.

[29] Anthropic. Claude sonnet 4.6. `https://www.anthropic.com/news/claude-sonnet-4-6`, 2026.

[30] OpenAI. introducing-gpt-5. `https://openai.com/index/introducing-gpt-5/`, 2025.

[31] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[32] Qwen Team. Qwen3.6-Plus: Towards real world agents, April 2026. URL `https://qwen.ai/blog?id=qwen3.6`.

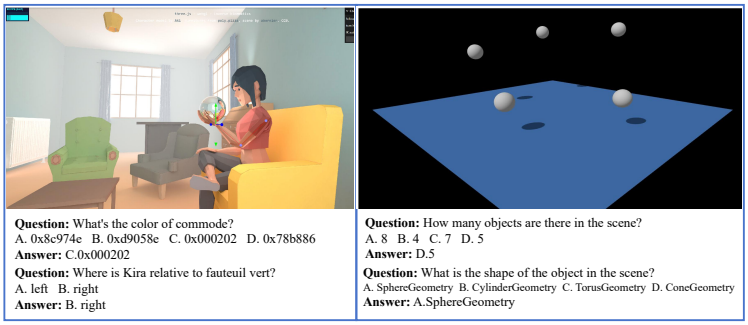

Figure 4: Data case.

A Technical appendices and supplementary material

A.1 Data Examples

Some simple data case is shown in Figure 4. In the QA example of Scene 1, Question 1 is about the color of the commode and belongs to the color recognition task in basic visual attributes. It requires the MLLM to first identify the commode in the scene and then further recognize its color. Question 2 is about judging the positional relationship of a person with the fauteuil vert from this perspective, which requires the MLLM to identify any person and the fauteuil vert in the scene and then further determine their positional relationship. In Scene 2, Question 1 is about counting the number of objects, and Question 2 belongs to the shape recognition task in basic visual attributes.

For each question, the answer options are generated by collecting all possible answers of the same type and then randomly sampling and shuffling them to form the final choices.

A.2 Further Analysis of Performance

To gain deeper insights into each evaluation dimension, we categorize the questions by their primary inquiry focus. The resulting fine-grained accuracy metrics are reported in Table 3 in the main paper.

Across static tasks, models achieve consistently high accuracy in shape recognition, with an average accuracy of 88.75%, and multiple models reaching 100%. This result indicates that basic geometric perception is well learned by current MLLMs. In comparison, color recognition shows larger variance (average 56.34%), suggesting sensitivity to visual noise and style variations in web-based runtime environments.

For counting tasks, models achieve moderate performance (average 58.77%), with top models such as GPT-5.4 and Seed-2.0-Mini performing well, while others exhibit substantial degradation. The result suggests that numerical reasoning over spatial layouts remains unstable, particularly in the presence of cluttered or occluded scenes.

Spatial relation reasoning is relatively well handled by most models, with GPT-5.4 achieving the highest accuracy (95.65%). However, performance drops noticeably in more complex settings involving multiple objects or ambiguous viewpoints, indicating limited robustness in compositional reasoning.

For dynamic tasks, models perform reasonably well on direction transformation, with several models exceeding 70% accuracy, including Qwen3 VL 32B Instruct and Claude Sonnet 4.6. However, models that perform strongly on static tasks, such as the GPT series, do not exhibit the same advantage here, suggesting a gap between visual perception and dynamic spatial reasoning.

Rotation transformation emerges as the most challenging task type. Most models perform poorly, with some achieving accuracy as low as 11.11%. Even the best-performing models, such as Qwen3 VL 32B Instruct and Qwen3 VL 8B Thinking, reach only 60.49%. It indicates that rotation reasoning requires more advanced spatial abstraction and mental transformation capabilities, which remain a major limitation of current MLLMs.

Overall, these results highlight that while current models are effective at basic spatial perception and simple relation reasoning, they struggle with tasks requiring compositional reasoning and dynamic spatial transformations.

A.3 Code and Reproducibility

To facilitate reproducibility, we publicly release all scripts and tools used for data generation and model evaluation at `https://github.com/ShaofeiLiu-sofia/WebSpatial/`. The repository contains (1) scripts for runtime data extraction and scene parsing, (2) the QA-generation pipeline, and (3) evaluation and analysis scripts.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: See in Section 1.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See in Section 5.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: NA.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: See in section 4 and appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: See in Appendix.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: See in section 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: See in section 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [N/A]

   Justification: NA.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: NA.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: NA.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: NA.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: These have been cited in the paper.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No]

    Justification: NA.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: NA.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: NA.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: See in Section 3.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.