Scenethesis:
Structure-Based XR Scene Synthesis

LYU2406

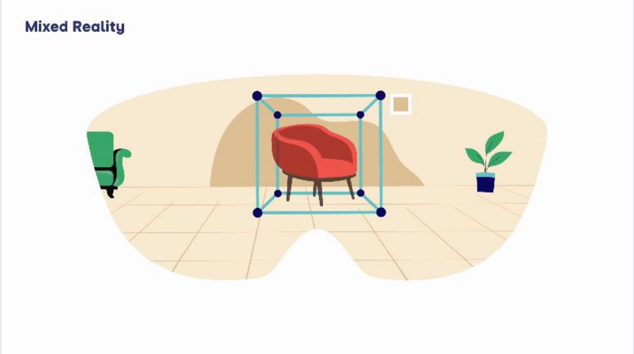
Supervisor: Professor Michael R. Lyu

Presenter: LAM Yiu Fung Anson

Outline

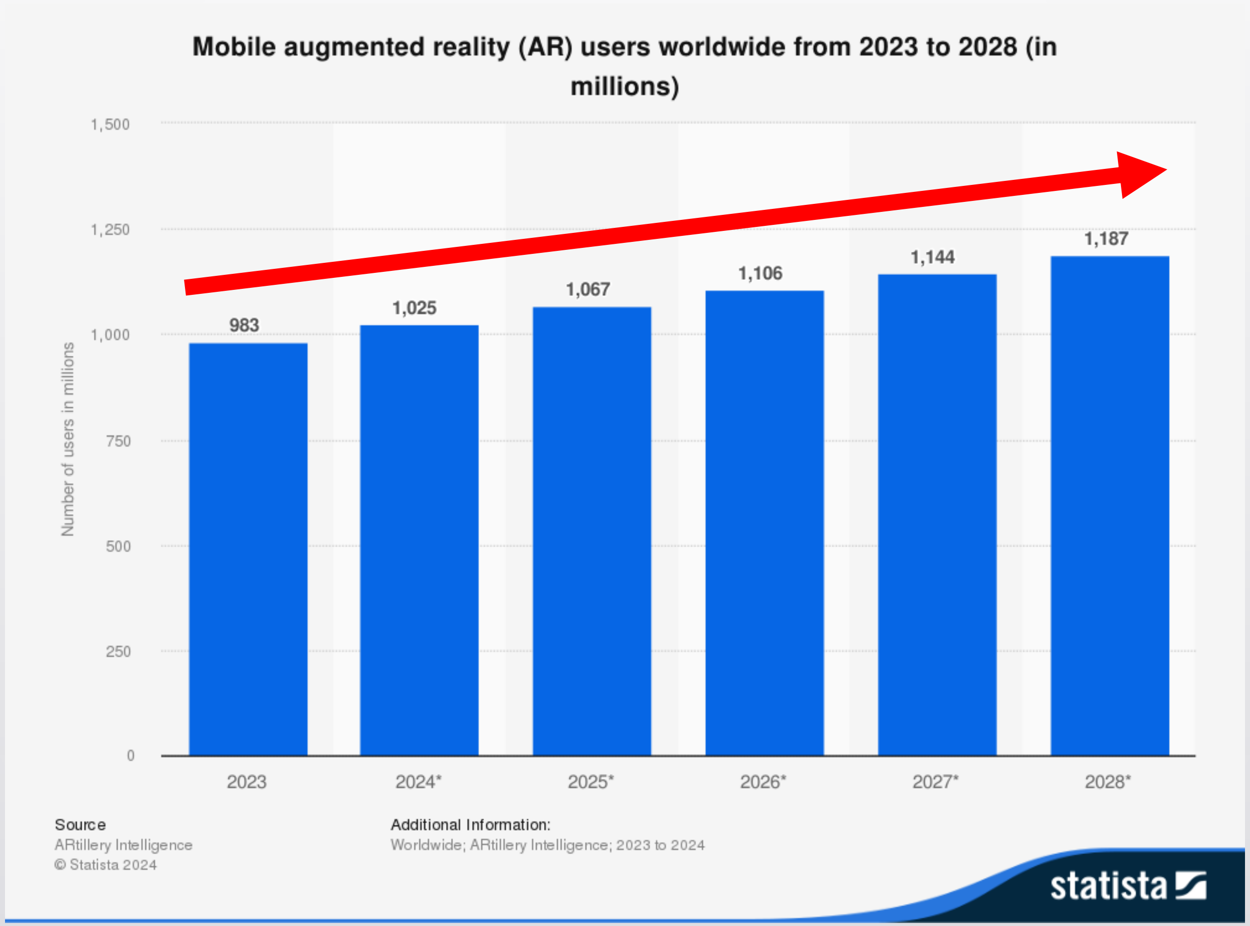
1. Background
2. Motivation
3. Methodology
4. Demo
5. Experiments
6. Conclusion
7. Future Work

Background

Extended Reality (XR)		
Virtual Reality (VR)	**Augmented Reality (AR)**	**Mixed Reality (MR)**
Brings users to a fully simulated and isolated world	Combines the real and virtual worlds by overlaying digital content onto a dedicated device	Similar to AR, but things happening in the physical world can affect the virtual world
		

Video source: The differences between AR, VR & MR (https://www.youtube.com/watch?v=IFgGzOpjlUM)

Background


Mobile augmented reality (AR) users worldwide from 2023 to 2028 (in millions)
Number of users in millions
1,500
1,250
1,000
750
500
250
0
983
1,025
1,067
1,106
1,144
1,187
2023
2024*
2025*
2026*
2027*
2028*
Source
ARtillery Intelligence
© Statista 2024
Additional Information:
Worldwide; ARtillery Intelligence; 2023 to 2024
statista

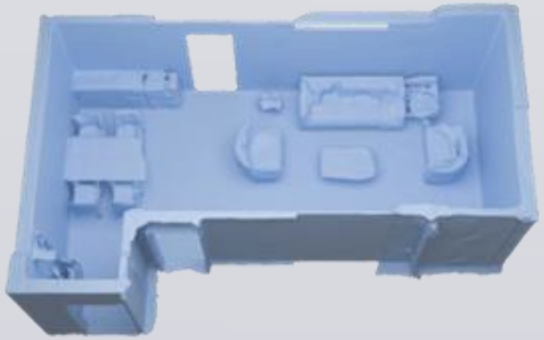
Motivation

Software testing	
Non-XR software	**XR (AR and MR) software**
1. Use DFS to go through every function 2. Perform unit tests automatically 😍😍😍	1. Design and construct various physical environments 2. Test all functions within each environment 3. Analyze the recordings manually to check for guideline violation 🥵🥵🥵

Term 1

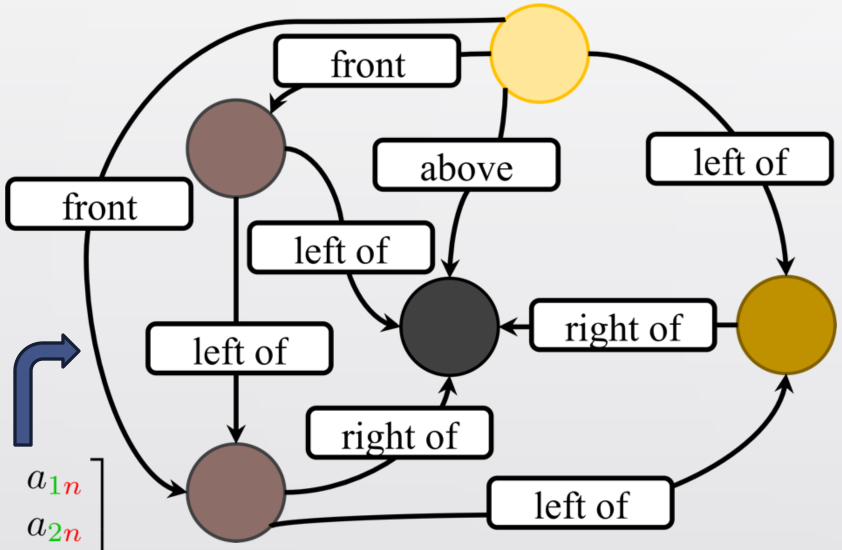
Can we also automate XR testing?

Motivation (by comparing related work)

Non-playable scene	Playable scene
A unified mesh → not physically correct  Text2Room, ICCV 2023  BlockFusion, SIGGRAPH 2024	Interactable with individual objects  InstructScene, ICCV 2024  Holodeck, CVPR 2024

Motivation (by comparing related work)

- Scene graph: node as object, edge as relation



$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ a_{31} & a_{32} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

- Categorical nodes, $o \in \{1, \ldots, K_o\}$
- Categorical edges, $r \in \{1, \ldots, K_r\}$

K_o, K_r are predefined!

1. Scene-specific, not flexible and practical for XR testing
2. Cannot accurately describe all relations

Image source: InstructScene

Motivation (by comparing related work)

We need a structure that is:

1. Human-understandable: for explainability and modifiability
2. Unambiguous: all (physically) possible relations in a scene can be modeled
3. Flexible: able to generate a diverse set of scenes

Methodology – ScenethesisLang

- Domain-specific language (DSL)
- Describes diverse, realistic, and physically plausible 3D scenes
- Focuses on expressiveness, human-readability, generative capability, and physical plausibility

Methodology – ScenethesisLang

$$
\begin{aligned}
scene \in Scenes ::= \ & \textbf{SceneType: } scene\_type; \\
& \textbf{SceneDescription: } scene\_description; \\
& \textbf{Regions: } regions; \\
& \textbf{Connections: } connections; \\
& \textbf{Constraints: } constraints; \\
& \textbf{ProbabilisticParameters: } probabilistic\_parameters; \\
& \textbf{TemporalRequirements: } temporal\_requirements
\end{aligned}
$$

Methodology – ScenethesisLang

- Scene type: overall nature of the environment (indoor/outdoor)
- Scene description: textual description of the scene

SceneType: $scene\_type$;

SceneDescription: $scene\_description$;

Regions: $regions$;

Connections: $connections$;

Constraints: $constraints$;

ProbabilisticParameters: $probabilistic\_parameters$;

TemporalRequirements: $temporal\_requirements$

Methodology – ScenethesisLang

- Region: spatial subdivision within a scene (e.g., room, outdoor area)
  - An outdoor area is defined as the bottom shape (from bird's-eye view) of a prism with infinite height.

$$
\begin{aligned}
regions \in Regions &::= region \mid region;\ regions \\
region \in Region &::= id \leftarrow \textbf{region(}description, shape, materials, \\
&\quad\ construction, objects, visibility, constraints\textbf{)} \\
shape \in Shape &::= cuboid \mid ellipsoid \mid mesh \mid \\
&\quad\ shape \cup shape \mid shape \cap shape \mid shape - shape \\
cuboid \in Cuboid &::= \textbf{cuboid(}corner.min, corner.max\textbf{)} \\
ellipsoid \in Ellipsoid &::= \textbf{ellipsoid(}center, a, b, c\textbf{)} \\
mesh \in Mesh &::= \textbf{mesh(}vertices, faces\textbf{)} \\
corner \in Corner &::= \textbf{min} \mid \textbf{max} \\
materials \in Materials &::= floor\_material;\ non\_floor\_material \\
floor\_material \in Material &::= \textbf{material(}color, texture, reflectivity, opacity\textbf{)} \\
non\_floor\_material \in Material &::= \textbf{material(}color, texture, reflectivity, opacity\textbf{)}
\end{aligned}
$$

$$
\begin{aligned}
construction \in Construction &::= vertices, faces \\
vertices \in Vertices &::= [(x, y, z), ...] \mid [(x, y), ...] \\
faces \in Faces &::= [(index1, index2, index3), ...]
\end{aligned}
$$

Wall? Ceiling?

SceneType: *scene_type*;

SceneDescription: *scene_description*;

Regions: *regions*;

Connections: *connections*;

Constraints: *constraints*;

ProbabilisticParameters: *probabilistic_parameters*;

TemporalRequirements: *temporal_requirements*

Methodology – ScenethesisLang

- Object: entity within a region

$$
\begin{aligned}
objects \in Objects &::= object \mid object;\ objects \\
object \in Object &::= id \leftarrow \mathbf{object(}category, description, dimensions, \\
&\quad position, rotation, bounding\_box, \\
&\quad probabilistic\_properties, visibility\mathbf{)} \\
category \in Categories &::= \mathbf{text} \\
description \in ObjectDescription &::= text \\
dimensions \in Dimensions &::= \mathbf{(length,\ width,\ height)} \\
position \in Position &::= \mathbf{(x,\ y,\ z)} \mid \mathbf{offset(position,\ vector)} \mid \\
&\quad \mathbf{relativeTo(reference)} \\
rotation \in Rotation &::= \mathbf{(roll,\ pitch,\ yaw)} \mid \mathbf{relativeTo(reference)} \\
bounding\_box \in BoundingBox &::= \mathbf{bounding\_box(}min, max\mathbf{)} \\
visibility \in Visibility &::= \mathbf{rayTrace(}density, occlusion\mathbf{)} \mid \\
&\quad \mathbf{regionVisibility(}region, conditions\mathbf{)}
\end{aligned}
$$

SceneType: *scene_type*;

SceneDescription: *scene_description*;

Regions: *regions*;

Connections: *connections*;

Constraints: *constraints*;

ProbabilisticParameters: *probabilistic_parameters*;

TemporalRequirements: *temporal_requirements*

Methodology – ScenethesisLang

- Object: entity within a region

$$
\begin{aligned}
lights \in \mathit{Lights} &::= light \mid light;\ lights \\
light \in \mathit{Light} &::= id \leftarrow \textbf{light(}category, description, intensity, position, visibility\textbf{)} \\
intensity \in \mathit{Intensity} &::= \textbf{float} \mid \textbf{distribution}
\end{aligned}
$$

SceneType: *scene_type*;

SceneDescription: *scene_description*;

Regions: *regions*;

Connections: *connections*;

Constraints: *constraints*;

ProbabilisticParameters: *probabilistic_parameters*;

TemporalRequirements: *temporal_requirements*

Methodology – ScenethesisLang

- Connection: spatial relationship between two regions

$$connections \in \mathit{Connections} ::= connection \mid connection;\ connections$$
$$connection \in \mathit{Connection} ::= \textbf{connection(}region1, region2, type, description\textbf{)}$$

SceneType: $scene\_type$;

SceneDescription: $scene\_description$;

Regions: $regions$;

Connections: $connections$;

Constraints: $constraints$;

ProbabilisticParameters: $probabilistic\_parameters$;

TemporalRequirements: $temporal\_requirements$

Methodology – ScenethesisLang

- Constraint: requirement that ensures physical plausibility or reasonableness, or that meets user-defined specifications

$$
\begin{aligned}
constraints \in \mathit{Constraints} &::= constraint \mid constraint;\ constraints \\
constraint \in \mathit{Constraint} &::= spatial\_condition \mid probabilistic\_condition \mid \\
&\quad\ temporal\_condition \mid object\_relation \mid \\
&\quad\ visibility\_condition \mid physical\_constraint \mid \\
&\quad\ user\_defined(logic, priority)
\end{aligned}
$$

$$
\begin{aligned}
spatial\_condition \in \mathit{SpatialCondition} ::= &\ \textbf{inside(region)} \mid \textbf{outside(region)} \mid \\
&\ \textbf{above(object, height)} \mid \\
&\ \textbf{below(object, height)} \mid \\
&\ \textbf{nearby(object, distance)} \mid \\
&\ \textbf{alignedWith(object, axis)} \mid \\
&\ \textbf{tangentTo(surface)} \mid \\
&\ \textbf{distanceBetween(object1, object2) == d}
\end{aligned}
$$

SceneType: *scene_type*;

SceneDescription: *scene_description*;

Regions: *regions*;

Connections: *connections*;

Constraints: *constraints*;

ProbabilisticParameters: *probabilistic_parameters*;

TemporalRequirements: *temporal_requirements*

$$
\begin{aligned}
visibility\_condition \in \mathit{VisibilityCondition} ::= &\ \textbf{canSee(observer, target)} \mid \\
&\ \textbf{occludes(object1, object2)} \mid \\
&\ \textbf{visibleInRegion(observer, region)} \mid \\
&\ \textbf{rayTraceValid(observer, target, density)} \\
physical\_constraint \in \mathit{PhysicalConstraint} ::= &\ \textbf{noCollision(object1, object2)} \mid \\
&\ \textbf{collisionFreeRegion(region)} \mid \\
&\ \textbf{stablePosition(object)} \mid \\
&\ \textbf{gravityAligned(object)} \\
user\_defined \in \mathit{UserDefinedConstraint} ::= &\ \textbf{customLogic(logicExpression, priority)}
\end{aligned}
$$

$$
\begin{aligned}
object\_relation \in \mathit{ObjectRelation} ::= &\ \textbf{relation(object1, object2, relation\_type)} \\
relation\_type \in \mathit{RelationTypes} ::= &\ \textbf{above} \mid \textbf{below} \mid \textbf{inside} \mid \textbf{outside} \mid \textbf{nearby} \mid \textbf{aligned} \mid \\
&\ \textbf{occludes} \mid \textbf{intersects}
\end{aligned}
$$

Methodology – ScenethesisLang

- Probabilistic parameter: ranges or distributions for attributes such as object positions

$$probabilistic\_condition \in ProbabilisticCondition ::=$$

$$\textbf{probability(p): condition} \mid \textbf{distributionBased(object, param: distribution)}$$

SceneType: $scene\_type$;

SceneDescription: $scene\_description$;

Regions: $regions$;

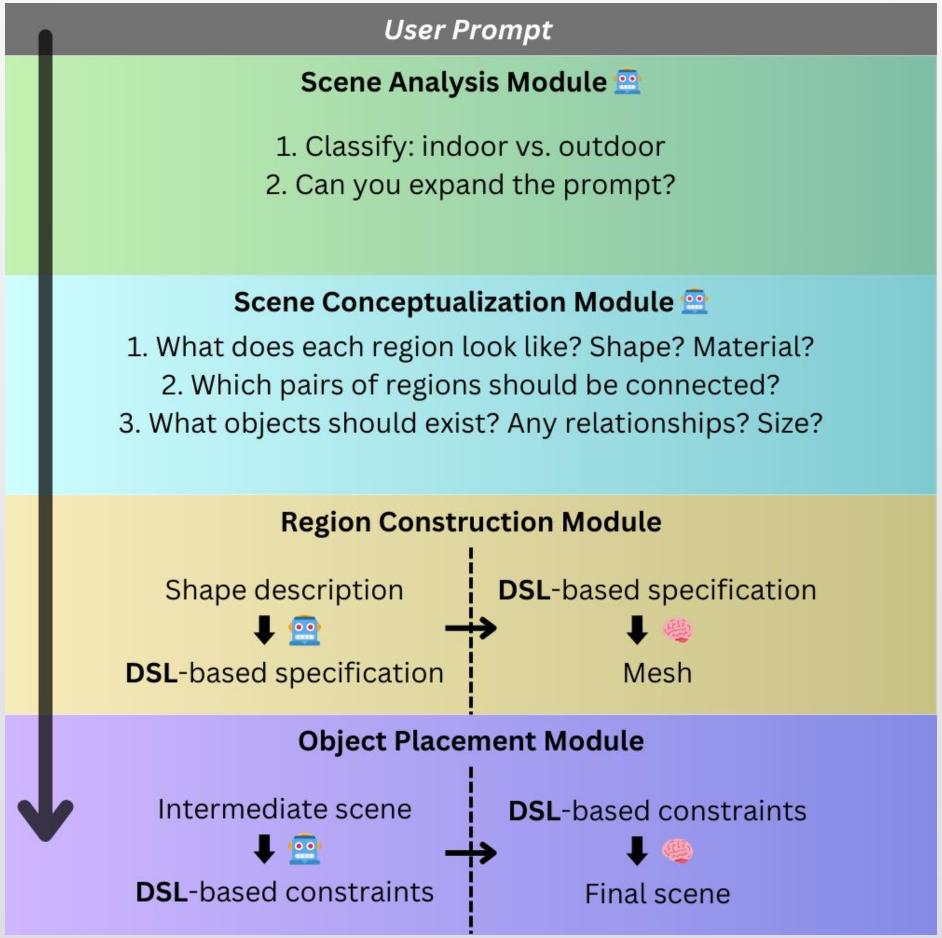
Connections: $connections$;

Constraints: $constraints$;

ProbabilisticParameters: $probabilistic\_parameters$;

TemporalRequirements: $temporal\_requirements$

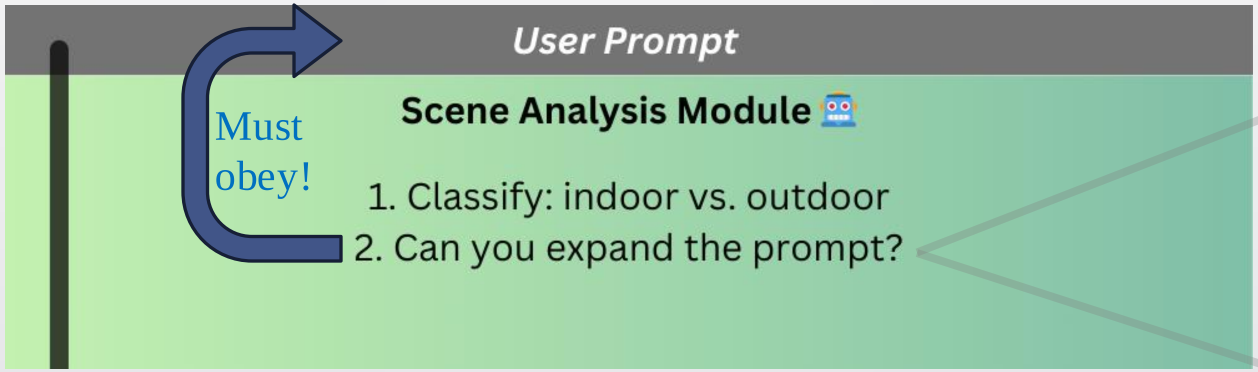
Methodology – Scenethesis



: LLM

Methodology – Scenethesis

- Scene Analysis Module: enhances the overall understanding of the desired scene



- Purpose of existence
- Example objects
- Expected human activities
- Overall atmosphere

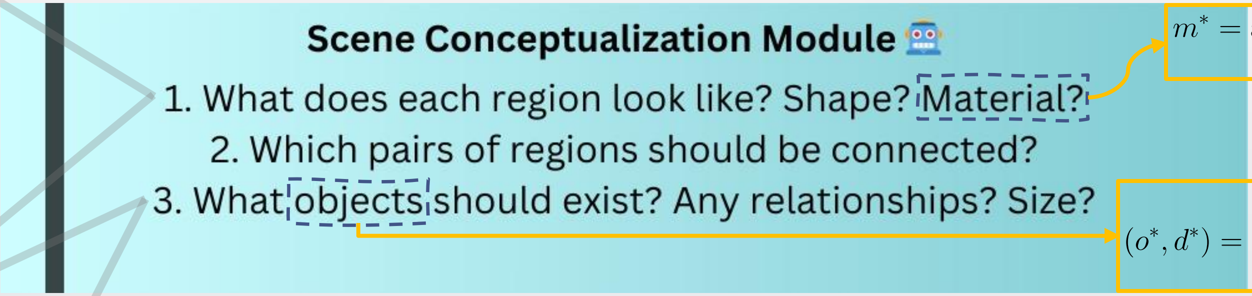
$$\begin{aligned} \mathit{scene} \in \mathit{Scenes} ::= {} & \textbf{SceneType:}\ \mathit{scene\_type}; \\ & \textbf{SceneDescription:}\ \mathit{scene\_description}; \end{aligned}$$

Methodology – Scenethesis

- Scene Conceptualization Module: creates a semantic draft of the desired scene
  - Semantic: only text is outputted

- Purpose of existence
- Example objects
- Expected human activities
- Overall atmosphere

3 prompts per region:
1. Names of all objects
2. Relationships
3. Attributes



$$m^* = \operatorname*{argmax}_{m \in \mathcal{D}_m} \text{CLIP}(m, q_m)$$

$$(o^*, d^*) = \operatorname*{argmax}_{(o,d) \in \mathcal{D}_o} \frac{\alpha \cdot \max_{r \in \mathcal{R}} \text{CLIP}(r, q_o) + \beta \cdot \text{SBERT}(d, q_o)}{\alpha + \beta}$$

q_o: A 3D object of a {category} named {name} ({description})

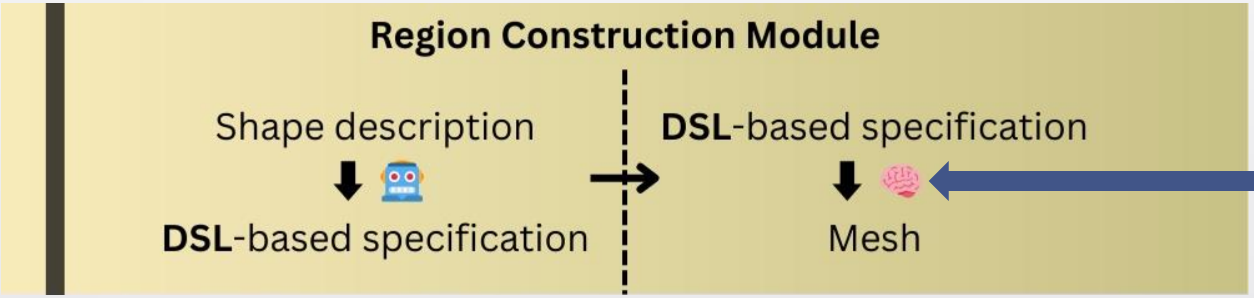
$$\mathit{region} \in \mathit{Region} ::= \mathit{id} \leftarrow \textbf{region(}\mathit{description}, \mathit{shape}, \mathit{materials}, \mathit{construction}, \mathit{objects}, \mathit{visibility}, \mathit{constraints}\textbf{)}$$

$$\mathit{connection} \in \mathit{Connection} ::= \textbf{connection(}\mathit{region1}, \mathit{region2}, \mathit{type}, \mathit{description}\textbf{)}$$

$$\mathit{object} \in \mathit{Object} ::= \mathit{id} \leftarrow \textbf{object(}\mathit{category}, \mathit{description}, \mathit{dimensions}, \mathit{position}, \mathit{rotation}, \mathit{bounding\_box}, \mathit{probabilistic\_properties}, \mathit{visibility}\textbf{)}$$

Methodology – Scenethesis

- Region Construction Module: constructs the actual boundaries of each region



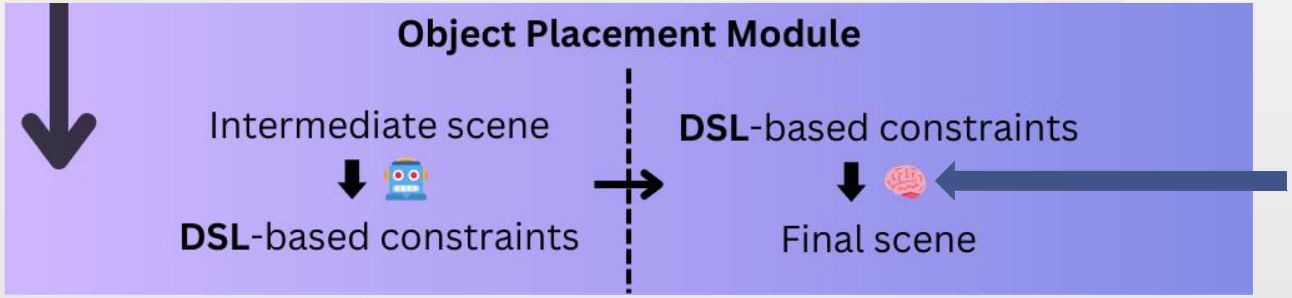
Shape parser

$$
\begin{aligned}
shape \in \mathit{Shape} &::= cuboid \mid ellipsoid \mid mesh \mid \\
&\quad\; shape \cup shape \mid shape \cap shape \mid shape - shape \\
cuboid \in \mathit{Cuboid} &::= \mathbf{cuboid(}corner.min, corner.max\mathbf{)} \\
ellipsoid \in \mathit{Ellipsoid} &::= \mathbf{ellipsoid(}center, a, b, c\mathbf{)} \\
mesh \in \mathit{Mesh} &::= \mathbf{mesh(}vertices, faces\mathbf{)} \\
corner \in \mathit{Corner} &::= \mathbf{min} \mid \mathbf{max}
\end{aligned}
$$

$$
\begin{aligned}
construction \in \mathit{Construction} &::= vertices, faces \\
vertices \in \mathit{Vertices} &::= \mathbf{[}(x, y, z), \ldots\mathbf{]} \mid \mathbf{[}(x, y), \ldots\mathbf{]} \\
faces \in \mathit{Faces} &::= \mathbf{[}(index1, index2, index3), \ldots\mathbf{]}
\end{aligned}
$$

Methodology – Scenethesis

- Object Placement Module: places selected objects to their optimal position with optimal orientation



Constraint solver

Demo

```
❯ genxr "A bedroom connected to a living room"
```

genxr 21:50:46

User prompt:
"A bedroom connected to a living room"

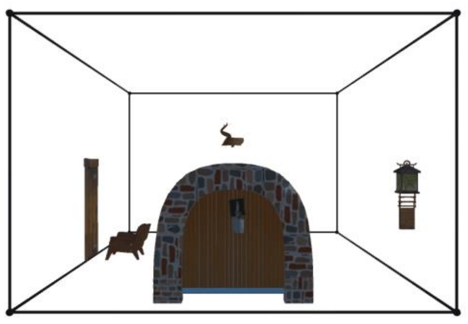
Experiments – Implementation

- Materials and objects are from Holodeck
- $\alpha = 100, \beta = 1$
- LLM: gpt-4o-2024-08-06
- Temperature: 0
- Machine: MacBook Pro (M1 Pro CPU, 16 GB RAM)
  - Mean execution time (before launching Blender and Unity): 2.5 minutes

Experiments – Evaluation Metrics

- Quantitative:
  - $\text{Score}_{\text{CLIP}}(r, q) = (\text{CLIP}(r, q) + 1) \times 50$
    - q_1: "an image of a vibrant indoor scene"
    - q_2: user prompt
    - q_3: generated scene description
  - $\text{Score}_{\text{SBERT}}(d, p) = (\text{SBERT}(d, p) + 1) \times 50$
    - d is the scene description, p is the user prompt

Wireframe rendering r:
Pitch $\in \{0, 30, 45, 60, 90\}$
Yaw $\in \{0, 15, \ldots, 330, 345\}$



Experiments – Evaluation Metrics

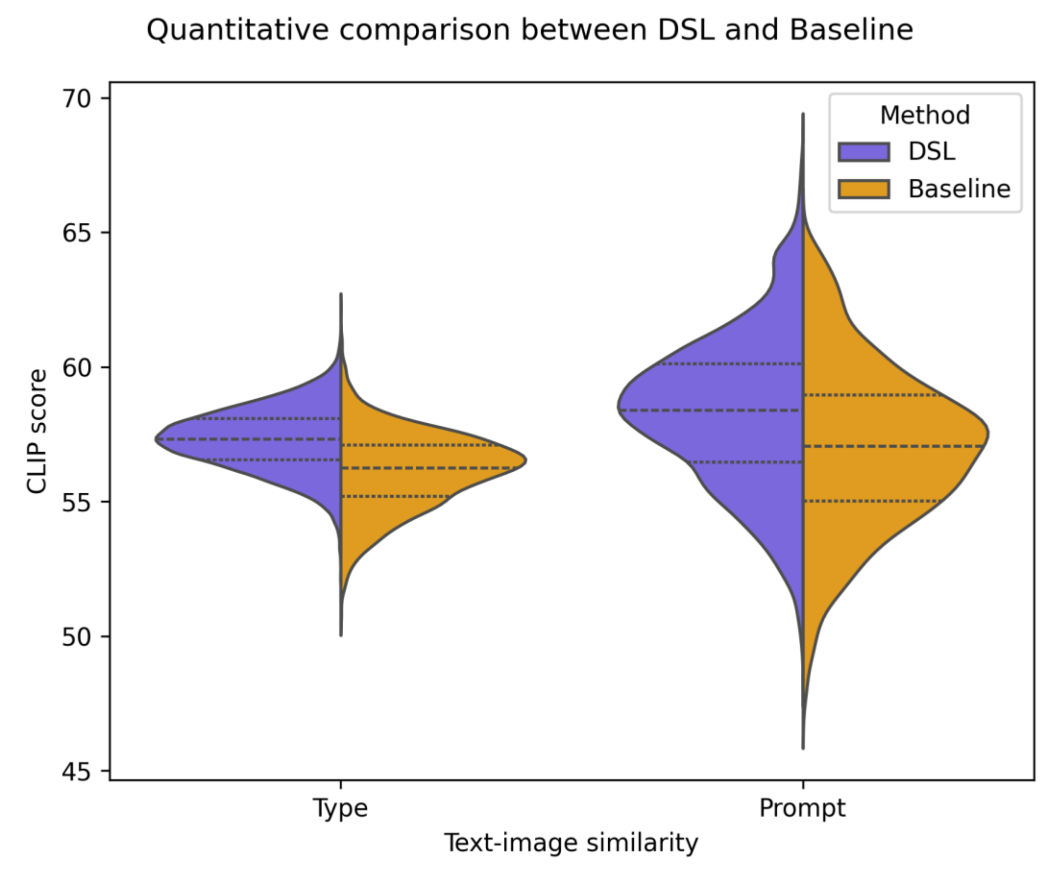
- Qualitative:
    - We give the bird's-eye view to GPT-4o and ask:
        1. Does the generated scene contain every region mentioned in the prompt?
        2. Does the generated scene contain every object mentioned in the prompt?
        3. Is the generated scene physically plausible (e.g., are there any objects colliding with region boundaries or other objects)?
        4. Is the generated scene visually pleasing?
    - Then rate from 1 to 10 with explanation

Experiments – Results

1. Comparing with baseline
    - Baseline: Non-DSL injected to obtain a minimal output
    - 50 generated prompts, 3 trials per prompt

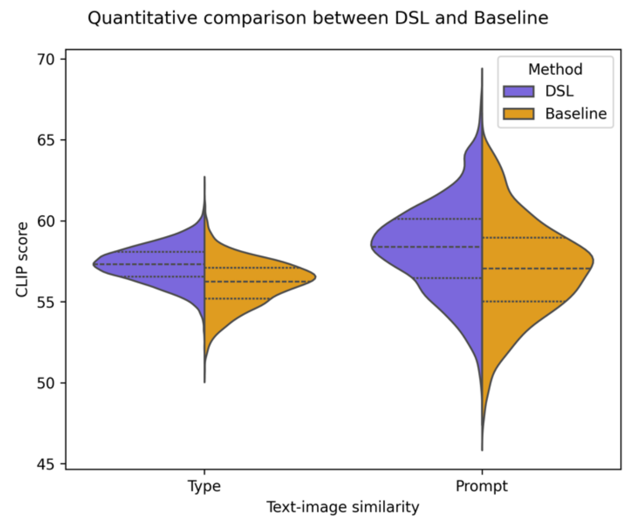
Experiments – Results

1. Comparing with baseline
    - Quantitative results:



Experiments – Results

1. Comparing with baseline
   - Quantitative results:
     - Small differences in CLIP scores
     - BUT! CLIP models were trained with real-life images, many of which have a main subject
     - Insensitiveness + small differences → Perhaps significant



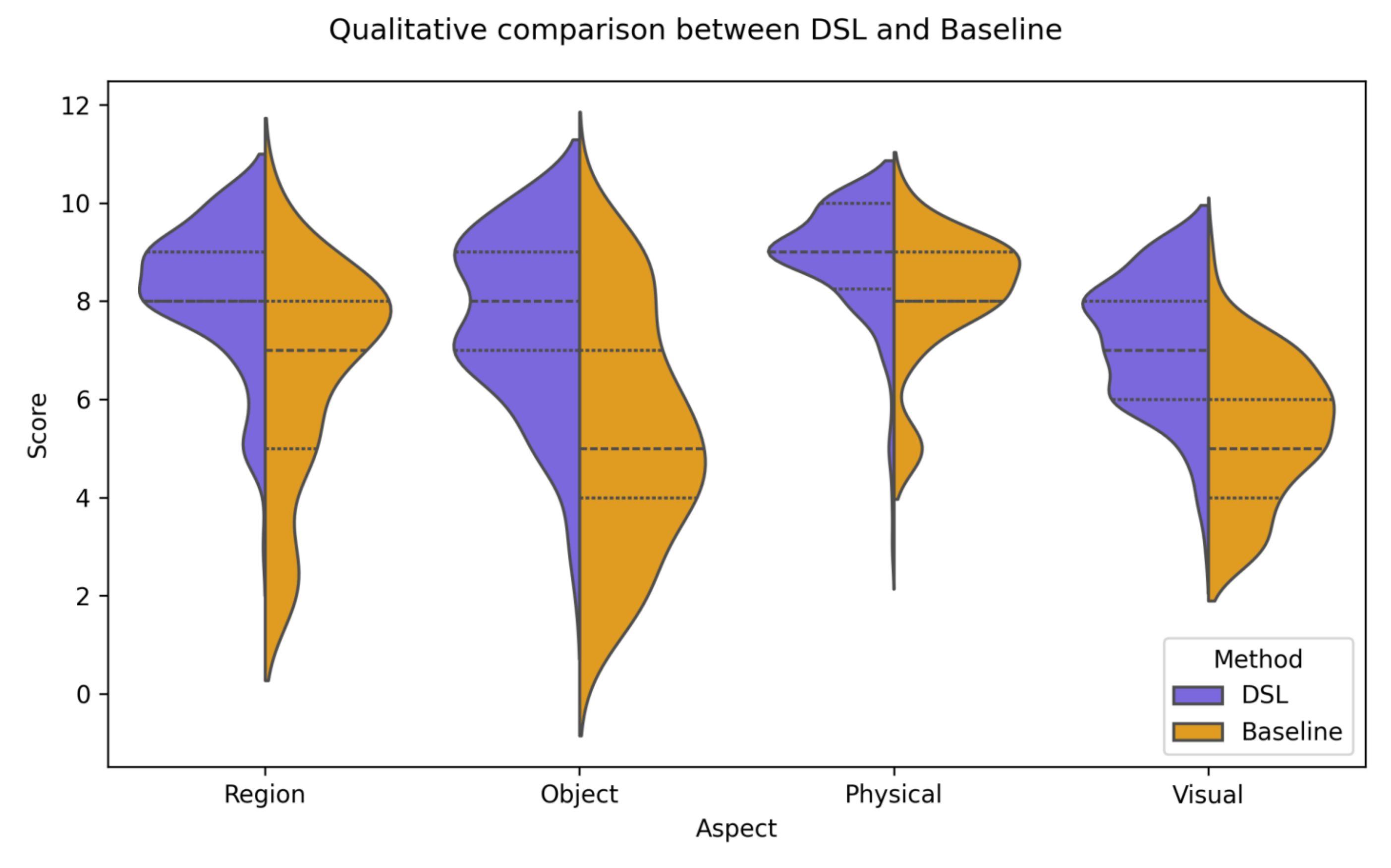
Experiments – Results

1. Comparing with baseline
   - Quantitative results:
     - Scenethesis generate significantly more objects → Vibrant and realistic

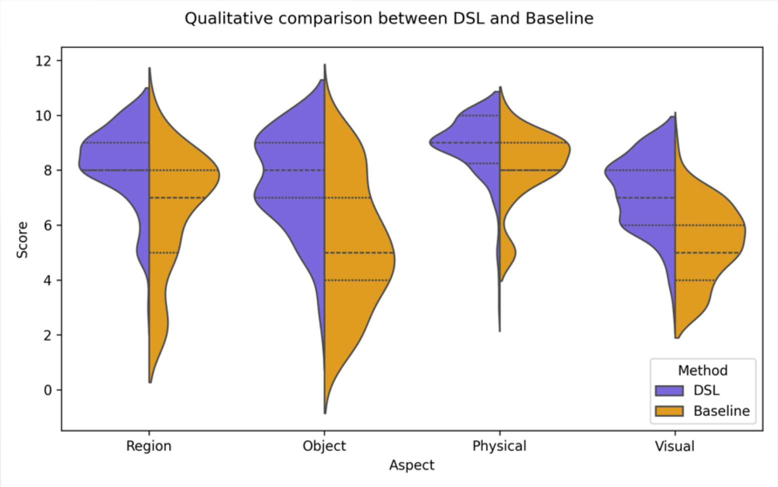
	Average # of regions	Average # of objects
Scenethesis	**1.36**	**20.973**
Baseline	1.35	3.63

Experiments – Results

1. Comparing with baseline
   - Qualitative results:



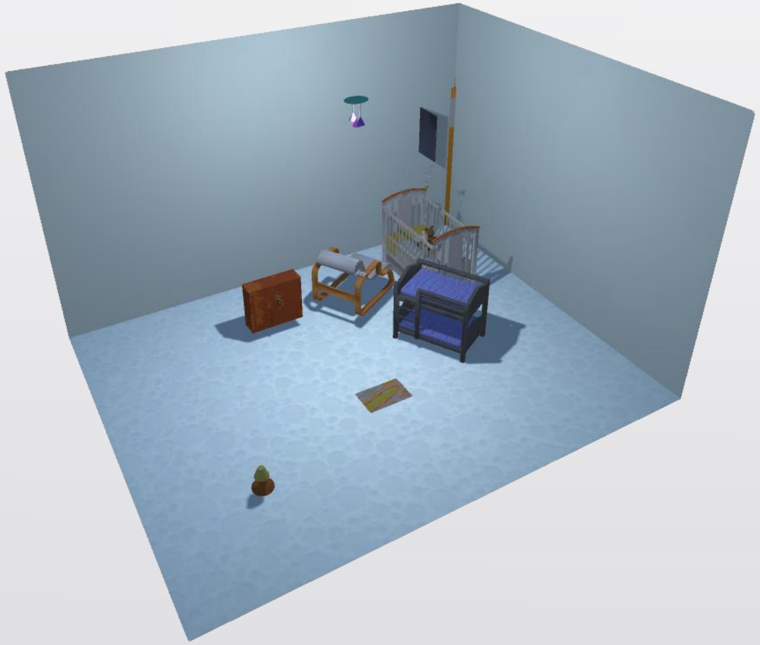
Experiments – Results

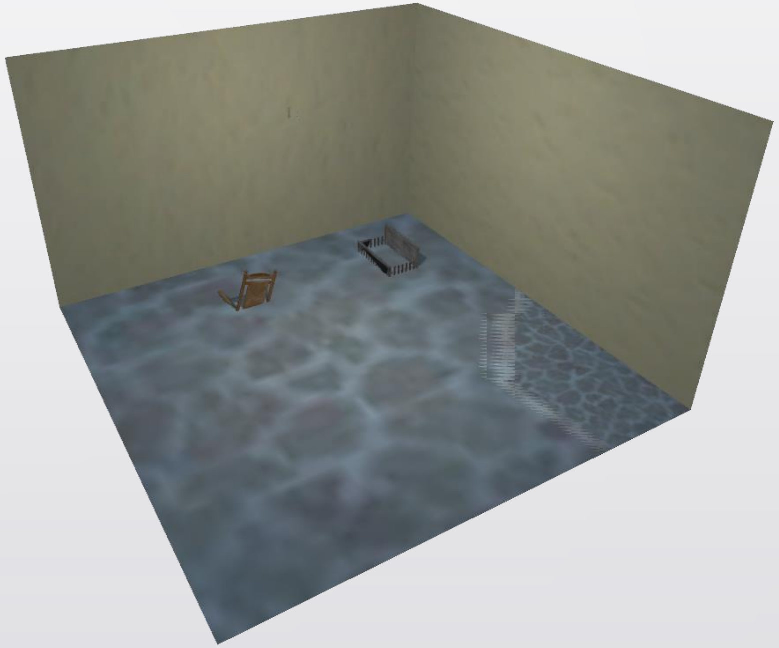


1. Comparing with baseline
   - Qualitative results:
     - Similar scores for “region” and “physical”
       - Scenethesis performs better in general
     - Significant performance gaps for “object” and “visual”
       - An evidence that Scenethesis can better follow user’s requirements while still producing visually pleasing scenes

Experiments – Results

1. Comparing with baseline
    - Examples (left is Scenethesis, right is baseline):

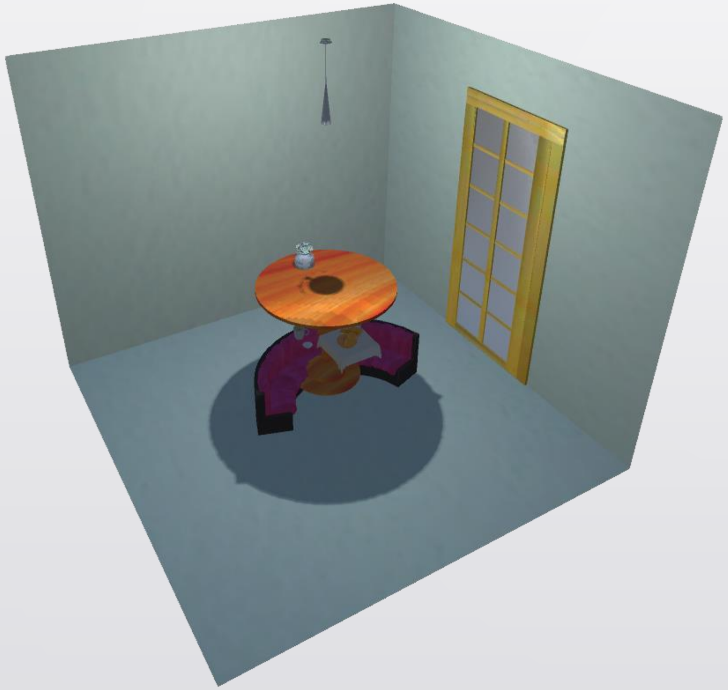


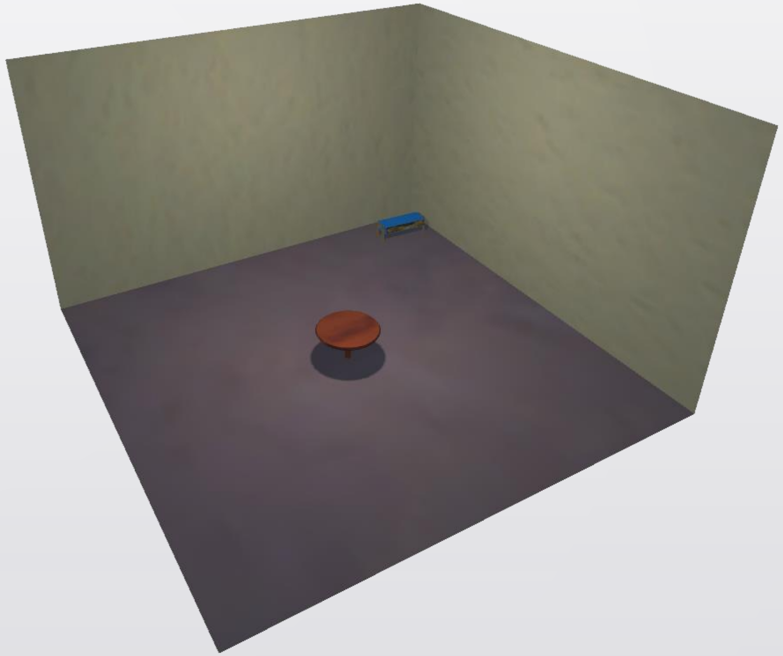


“A colorful nursery with a crib, rocking chair, and playful decor.”

Experiments – Results

1. Comparing with baseline
    - Examples (left is Scenethesis, right is baseline):





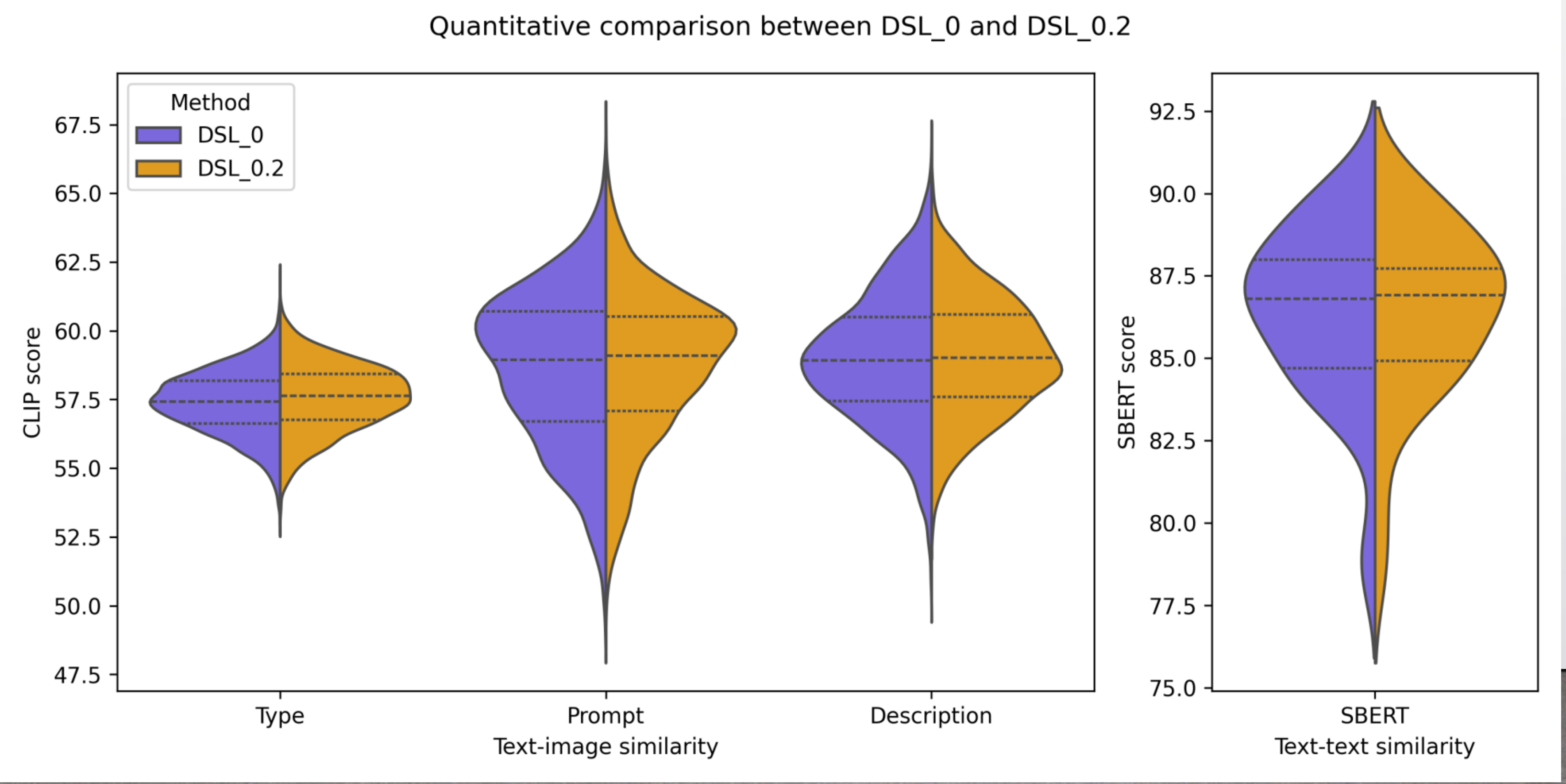
"A quaint breakfast nook with a round table and cushioned bench seating."

Experiments

2. Comparing different temperatures
   - $t \in \{0, 0.2, 0.5\}$
   - 20 generated prompts, 2 trials per prompt

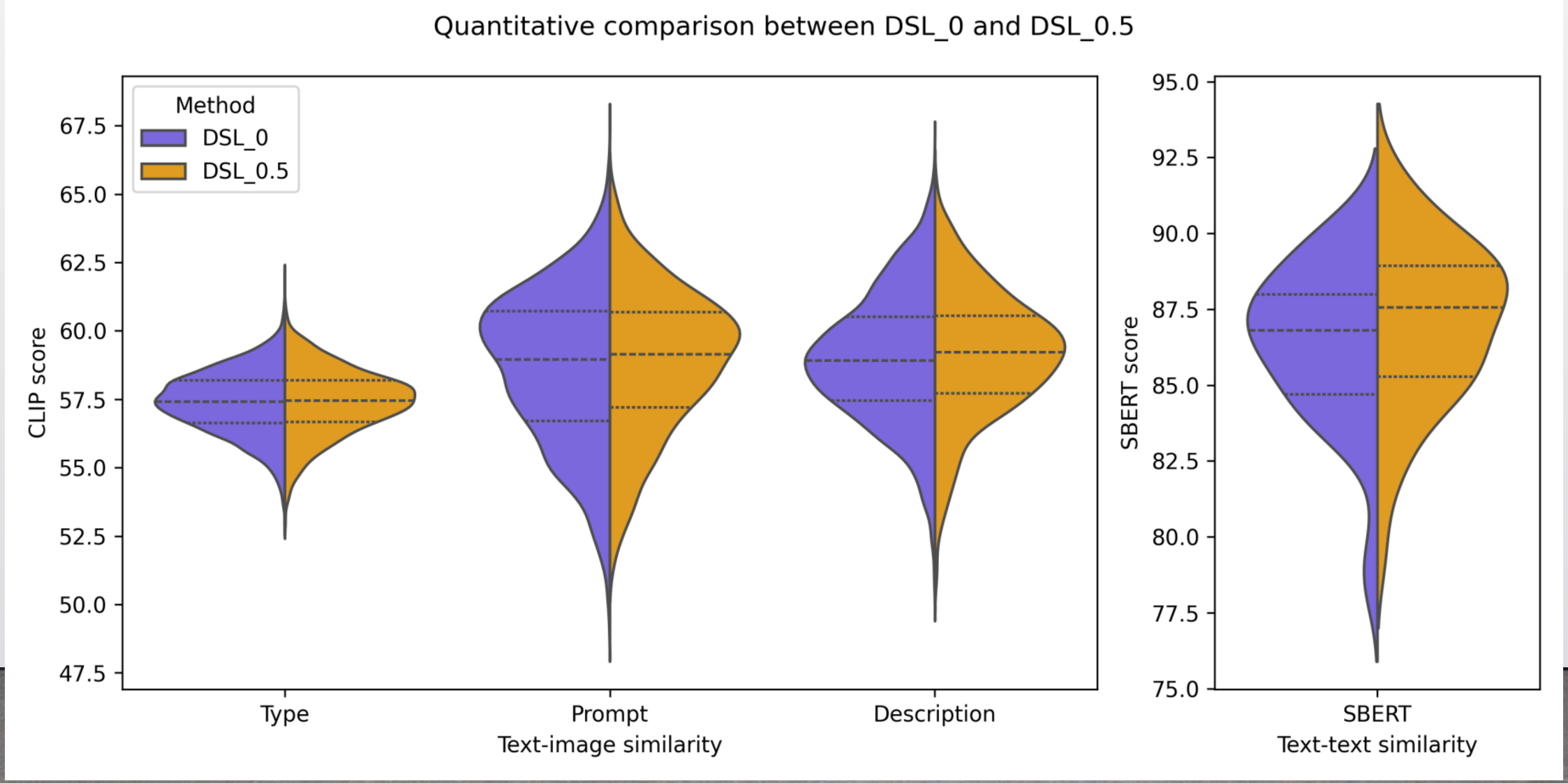
Experiments

2. Comparing different temperatures
   - Quantitative results: 0 vs 0.2



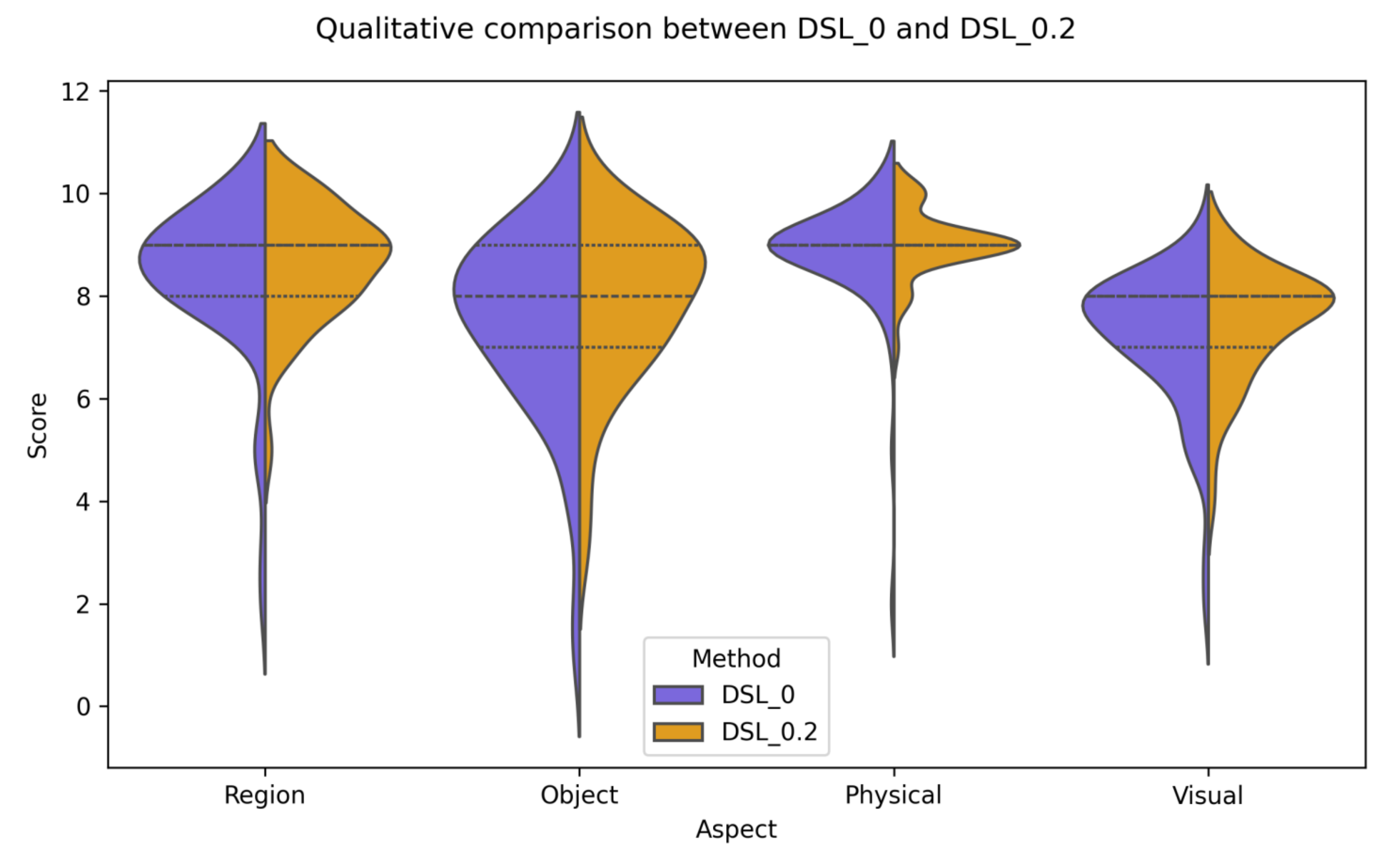
Experiments

2. Comparing different temperatures
   - Quantitative results: 0 vs 0.5



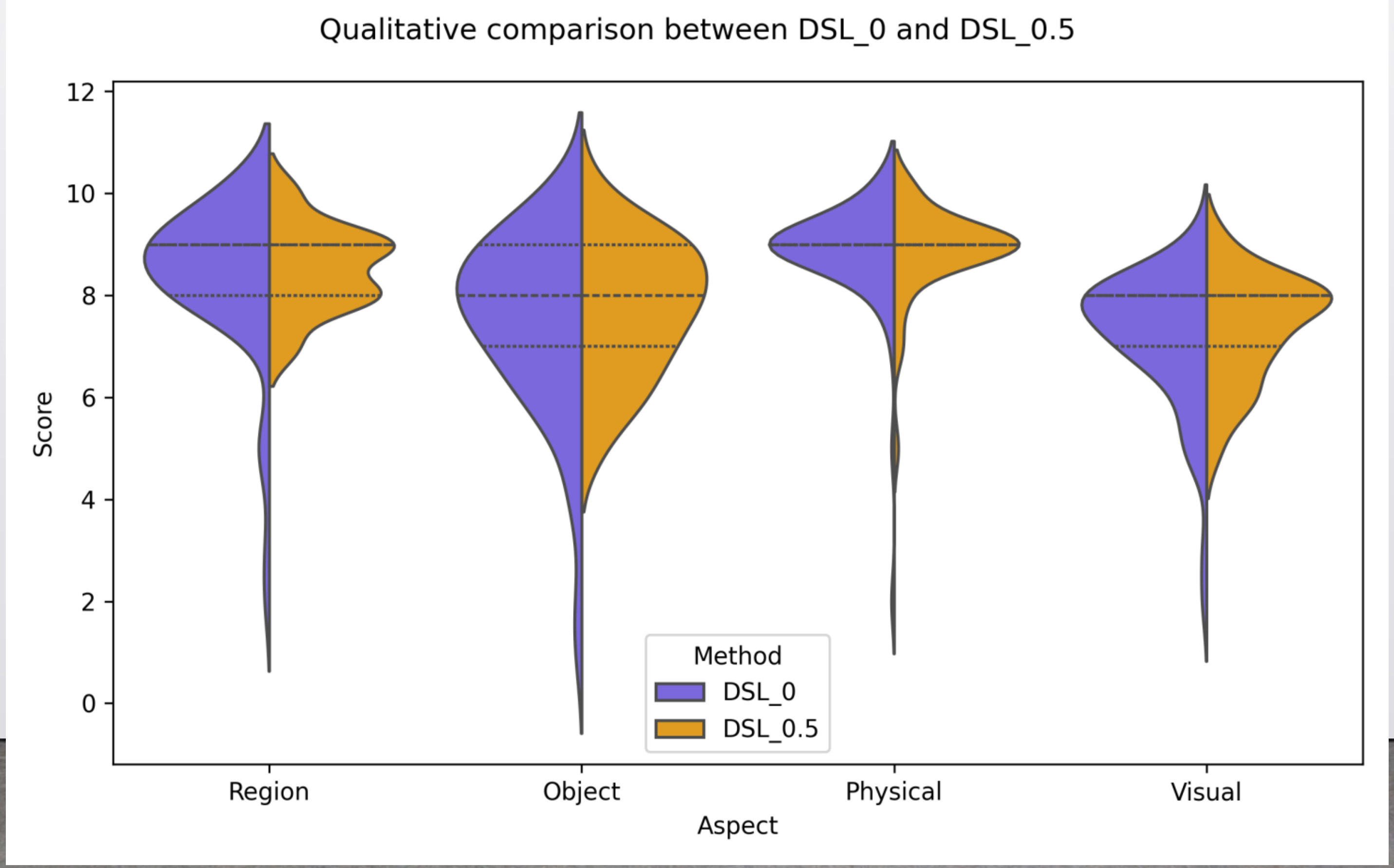
Experiments

2. Comparing different temperatures
   - Qualitative results: 0 vs 0.2



Experiments

2. Comparing different temperatures
    - Qualitative results: 0 vs 0.5



Experiments

2. Comparing different temperatures
    - Subjective visual difference is not significant
    - So, temperature is not a major factor on the performance of Scenethesis

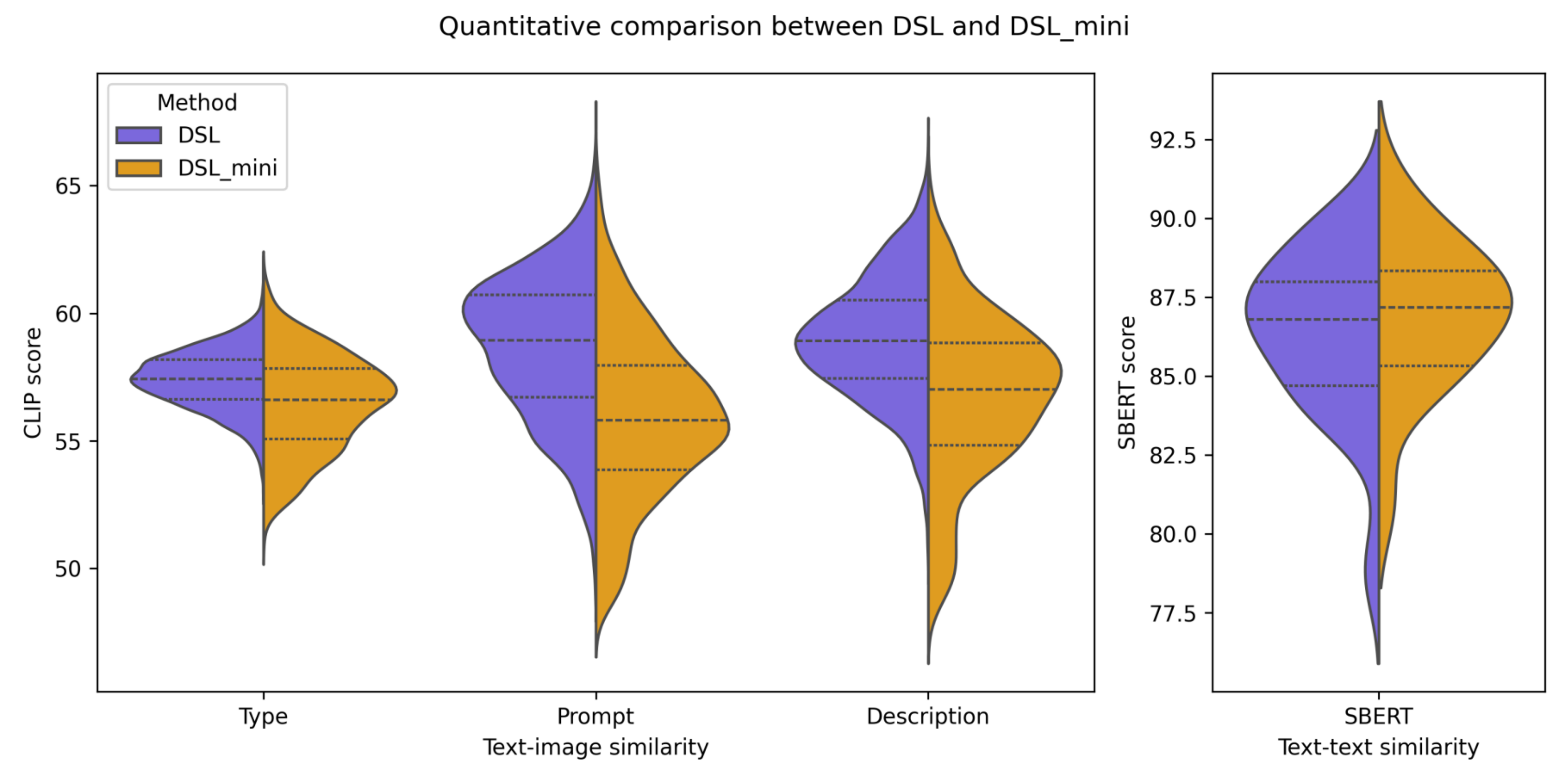
Experiments

3. Comparing different LLMs
   - Model: gpt-4o-2024-08-06, gpt-4o-mini-2024-07-18, claude-3-5-sonnet-20241022
   - 20 generated prompts (same as before), 2 trials per prompt

Experiments

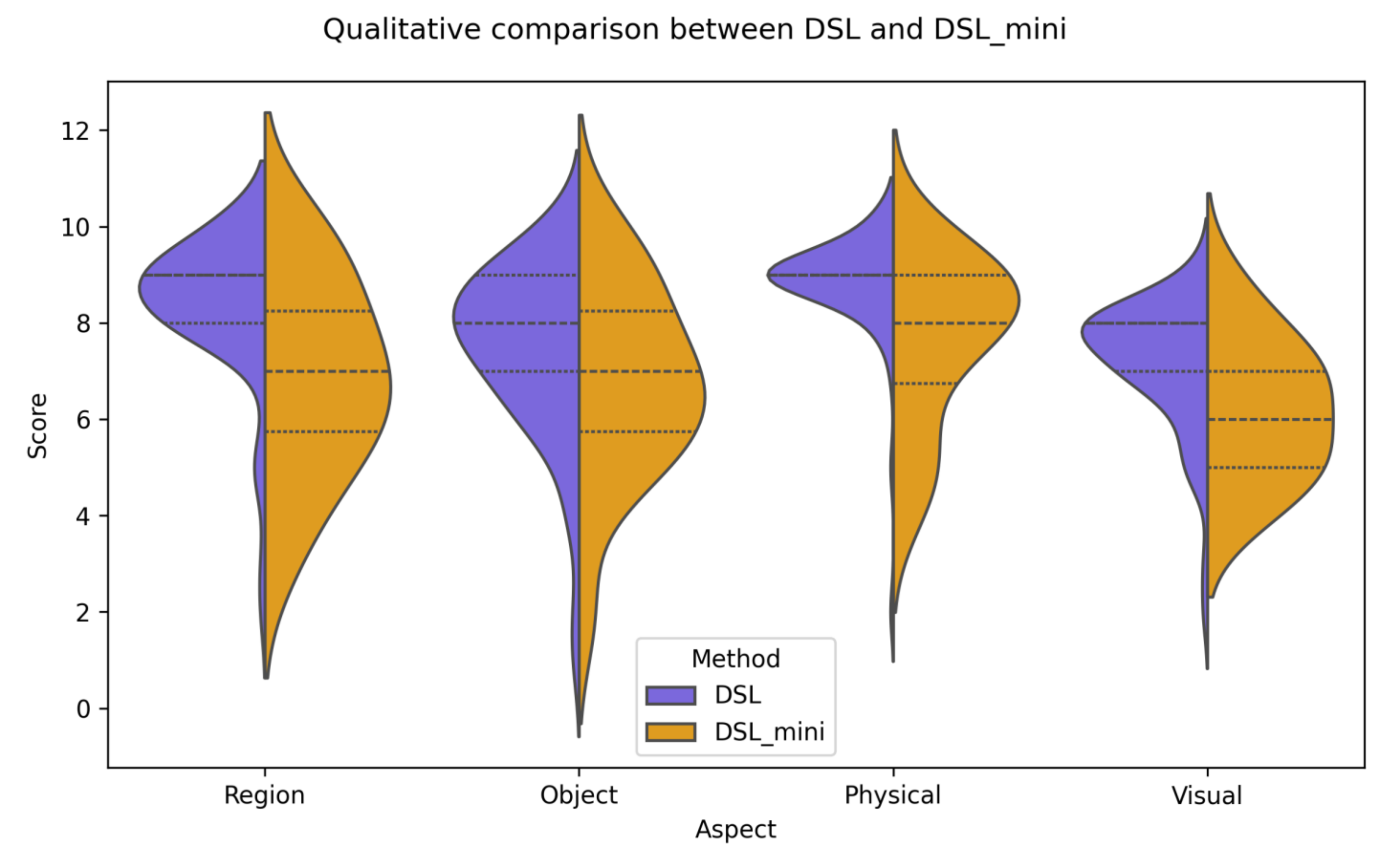
3. Comparing different LLMs

- Quantitative results: gpt-4o-mini-2024-07-18



Experiments

3. Comparing different LLMs
   - Qualitative results: gpt-4o-mini-2024-07-18

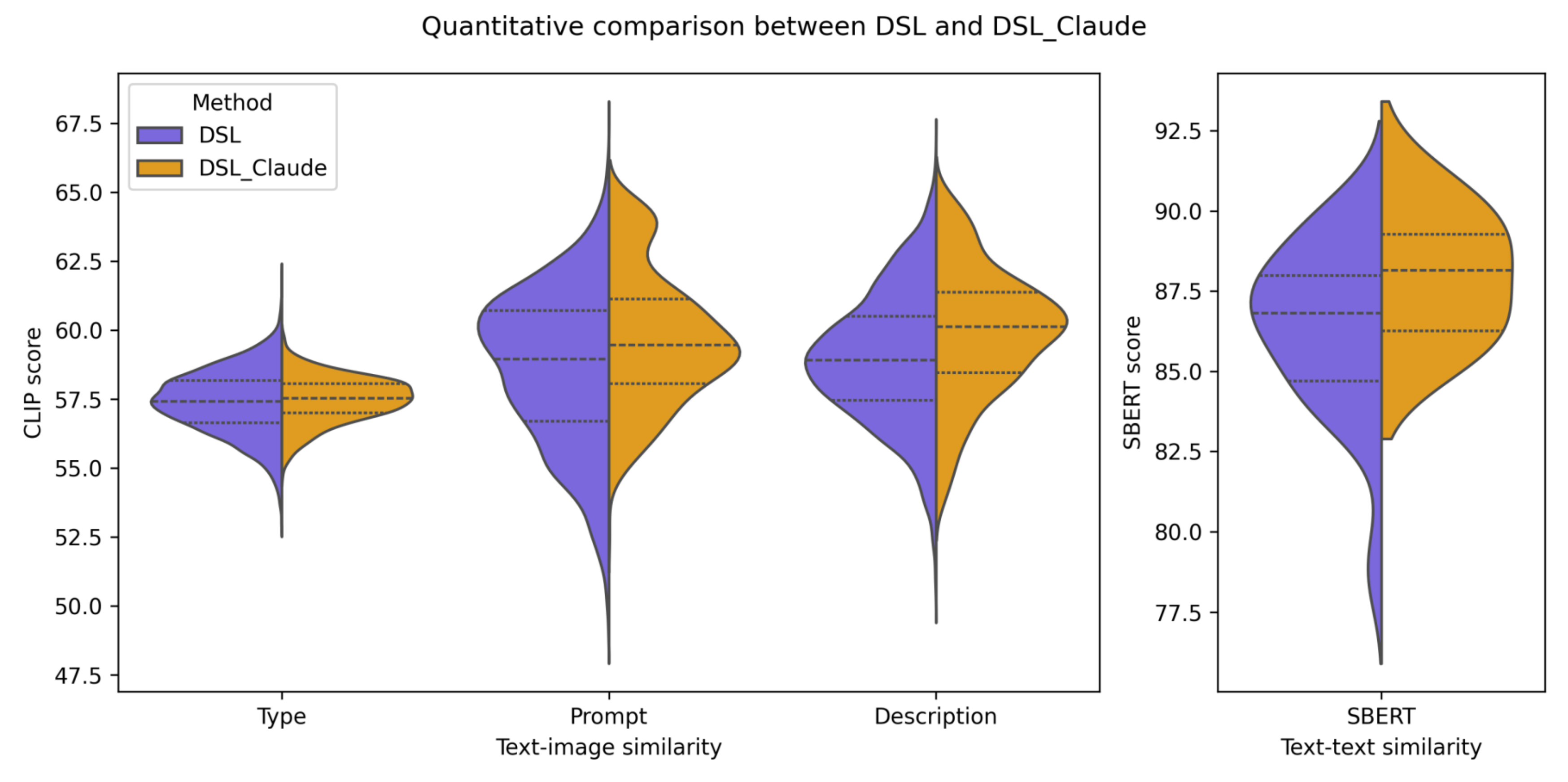


Experiments

3. Comparing different LLMs
   - As expected, larger models with better processing and generative capabilities work better

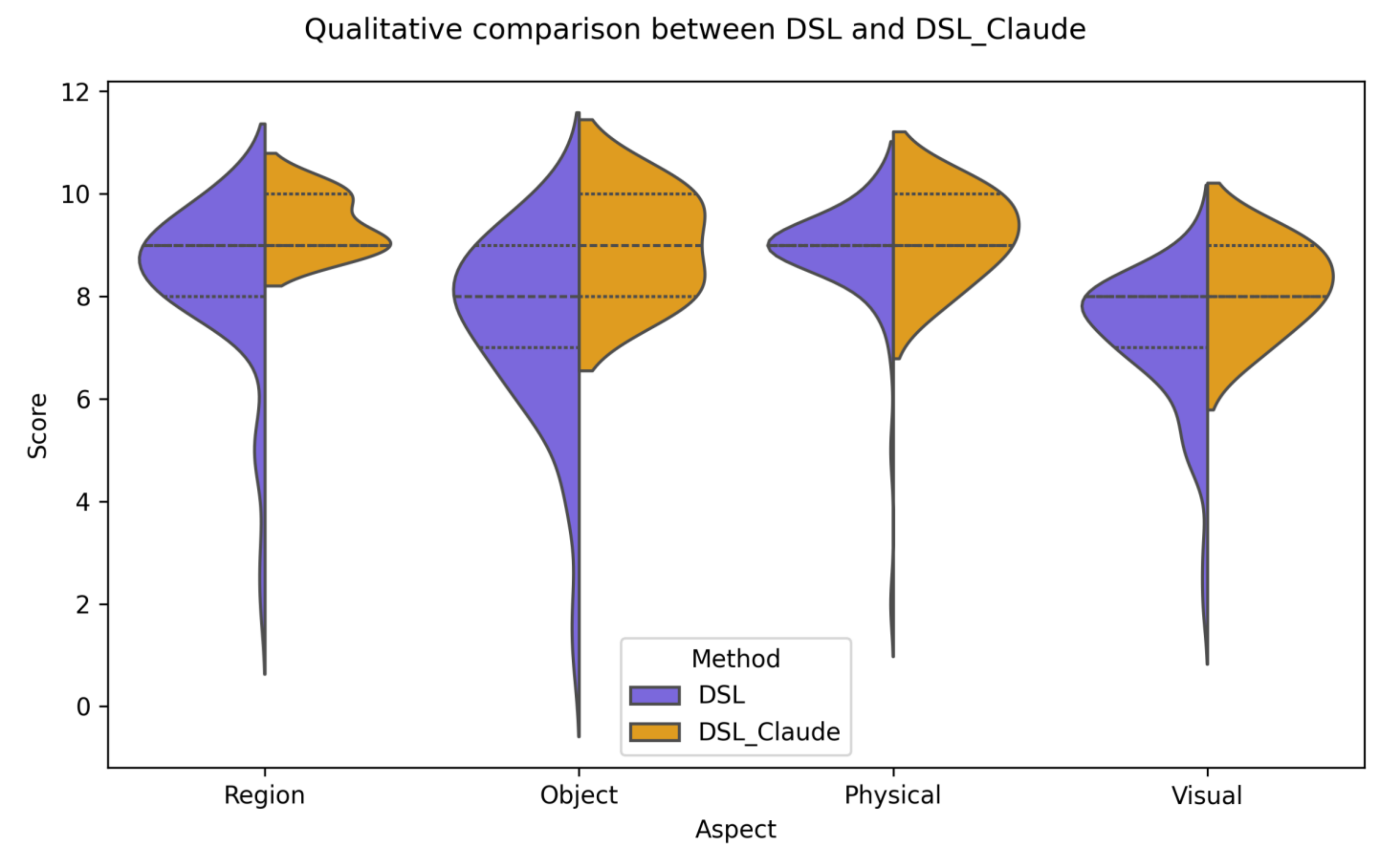
Experiments

3. Comparing different LLMs
   - Quantitative results: claude-3-5-sonnet-20241022 (only 5 scenes could be generated)



Experiments

3. Comparing different LLMs
   - Qualitative results: claude-3-5-sonnet-20241022 (only 5 scenes could be generated)



Experiments

3. Comparing different LLMs
    - An inspiration: models that are better at writing → better scene description → better scene

Conclusion

We propose:

1. ScenethesisLang, a novel domain-specific language designed for unambiguously describing a 3D scene; and
2. Scenethesis, a pipeline that takes a user prompt and generates a corresponding 3D scene using ScenethesisLang.

Experiments have demonstrated the potential of Scenethesis in generating vibrant and visually pleasing scenes.

Future Work

1. Reduce reliance on LLM
    - LLMs may not be able to handle numerical tasks accurately
    - Constraint solving is a highly complex task
    - Explore different shape parsers and constraint solvers

    → More realistic and physically plausible scene

Future Work

2. Employ object generation
    - Current curated database has only ~50K objects (~23 GB)
    - Original database has ~800K (~8 TB) → impractical to keep enlarging the database
    - With object generation model that can synthesize new objects in real time:
        - If the weighted object score is lower than a certain threshold → generate object
        - Acceptable to keep using a smaller database
        - No problem if the database does not have the target object

Future Work

3. Continue our journey on automatic XR testing
   - Scenethesis is just the first step in the final automated pipeline
   - With prompts tailor-made for specific target environments, we can generate a diverse set of environments
   - E.g., after figuring out how to make a smartphone recognize a virtual environment as a real one, we can freely test AR applications using our generated scenes
   - Then perhaps we can use a Vision Language Model to analyze key frames and produce an evaluation report

Q&A