Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

Bridging Language and 3D World Generation through Spatial-Graph Reasoning and Physics Constraints

Baibhab Das
School of Computer Science and Engineering, VIT-AP University, Andhra Pradesh, India.

Abstract - Teaching machines to reason about three-dimensional environments using natural language alone is genuinely hard. Existing text-to-3D methods routinely produce scenes that look physically absurd—objects float in mid-air, meshes intersect in impossible ways, and heavy objects rest atop flimsy supports. We tackle this head-on with a unified framework that weaves together natural language understanding, spatial reasoning, and physical plausibility. At its core sits a Spatially Grounded Transformer working alongside graph neural networks. Three components do the heavy lifting: a Graph Neural Scene Reasoner (GNSR) that explicitly models object relationships, a Latent Alignment Module (LAM) that bridges reasoning and generation, and a Physics-Aware Scene Consistency Module (PSCM) that catches physical nonsense. We push free-form text through an NLP pipeline that handles synonym replacement and dependency parsing, then feed the results into these modules. The system achieves 63.0% visual question answering accuracy across CLEVR, Objaverse, ShapeNet, and Text2Shape benchmarks. More importantly, it slashes the Physics Violation Score by 87.8%, dropping it from 41% to just 5%.

Keywords: Text-to-3D, Graph Neural Networks, Latent Alignment, Spatial Reasoning, Scene Understanding, Visual Question Answering, Physics-Aware Generation, Attention Mechanisms, Diffusion Models.

I. INTRODUCTION

Natural language processing and 3D scene understanding have each advanced considerably, but bringing them together remains stubbornly difficult. People describe visual scenes effortlessly—"the red cube sits to the left of the blue sphere"—yet building systems that can go the other direction, synthesizing or comprehending 3D scenes from such descriptions, opens exciting possibilities in robotics, augmented reality, and autonomous systems.

Here's the frustration: state-of-the-art text-to-image models like DALL-E and Stable Diffusion produce stunning 2D results, and single-object 3D generators like Shap-E and Point-E keep getting better. But throw multiple interacting objects at them, and the wheels come off. These systems don't reason about relationships between objects. They don't check whether a generated scene makes physical sense. The typical pipeline generates objects independently, then assembles them as an afterthought. The outcome? Floating objects, intersecting meshes, and scenes that violate basic physics.

What we really need is a system that understands object properties and physical constraints from the ground up. Not something that generates first and checks later, but something that builds physical plausibility into its core reasoning. That's exactly what we're proposing here.

Here's what we bring to the table:

1) **A multi-stage NLP parser** that extracts structured information from messy free-form text using regex, spaCy dependency parsing, synonym expansion, and sensible defaults when things get ambiguous.
2) **Graph Neural Scene Reasoner (GNSR)**: A hybrid architecture where objects become nodes in a graph and spatial relationships become edges—treating scene understanding as explicit relational reasoning rather than a black-box operation.
3) **Latent Alignment Module (LAM)**: A differentiable bridge that aligns our scene representations with Shap-E's latent space, making the whole pipeline end-to-end trainable.
4) **Physics-Aware Scene Consistency Module (PSCM)**: Three carefully designed loss functions that penalize collisions, unsupported objects, and gravitational impossibilities.
5) **Real-world validation**: Testing on Objaverse, ShapeNet, and Text2Shape alongside synthetic CLEVR data.
6) **Physics Violation Score (PVS)**: A new metric that quantifies just how physically plausible a generated scene really is.
7) **Thorough evaluation**: Comprehensive experiments and ablation studies that reveal what each component actually contributes.

Underneath all this lies what we call **Spatially-Grounded Relational Learning (SGRL)**. Three principles guide it: (1) build explicit relational graphs that capture object interactions, (2) let geometry inform attention mechanisms, and (3) keep reasoning and generation tightly coupled through differentiable alignment.

We've organized the paper like this. Section II surveys related work. Section III lays out the formal problem and mathematical machinery. Section IV walks through the architecture. Section V covers experiments and results.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

II. RELATED WORK

A. Text-to-Image and Text-to-3D Generation

Diffusion models have completely changed what's possible in text-to-image generation. Rombach et al. made a clever observation—running diffusion in compressed latent space rather than pixel space makes high-resolution synthesis tractable:

$$\mathcal{L}_{\text{LDM}} = \mathbb{E}_{z\sim\mathcal{E}(x),\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(z_t, t, c)\|_2^2\right] \tag{1}$$

Here $\mathcal{E}$ is the encoder, c is the text conditioning, and ϵ_θ is the denoising U-Net. Classifier-free guidance, introduced by Ho and Salimans, cranks up the influence of conditioning:

$$\tilde{\epsilon}_\theta(z_t, t, c) = \epsilon_\theta(z_t, t, \emptyset) + s \cdot (\epsilon_\theta(z_t, t, c) - \epsilon_\theta(z_t, t, \emptyset)) \tag{2}$$

with guidance scale $s > 1$.

On the 3D front, early voxel-based approaches hit a scaling wall—their $O(N^3)$ complexity made high-resolution outputs impractical. Neural radiance fields offered a more elegant path, representing scenes as continuous functions:

$$F_\theta : (\mathbf{x}, \mathbf{d}) \to (\mathbf{c}, \sigma) \tag{3}$$

where $\mathbf{x}$ is a 3D coordinate, $\mathbf{d}$ the viewing direction, $\mathbf{c}$ the predicted color, and σ the volume density. OpenAI's Shap-E went further, generating signed distances and colors conditioned on a latent code:

$$f_\theta : (\mathbf{x}, \mathbf{z}) \to (s, \mathbf{c}) \tag{4}$$

Here's the catch, though. All these systems are fundamentally single-object generators. They don't know how to arrange multiple objects in a scene. That's a different problem entirely, and it's the one we're tackling.

B. Scene Understanding and Reasoning

The CLEVR benchmark pushed VQA research forward by providing a controlled testbed for compositional reasoning. A CLEVR scene renders as:

$$\text{Scene} = \text{Render}(\mathcal{O}, \mathcal{R}, \mathcal{L}) \tag{5}$$

where $\mathcal{O}$ denotes objects, $\mathcal{R}$ spatial relations, and $\mathcal{L}$ lighting conditions. Figure 1 shows the exploratory analysis we ran on CLEVR—distributions of objects per scene, colors, shapes, materials, sizes, and question types.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

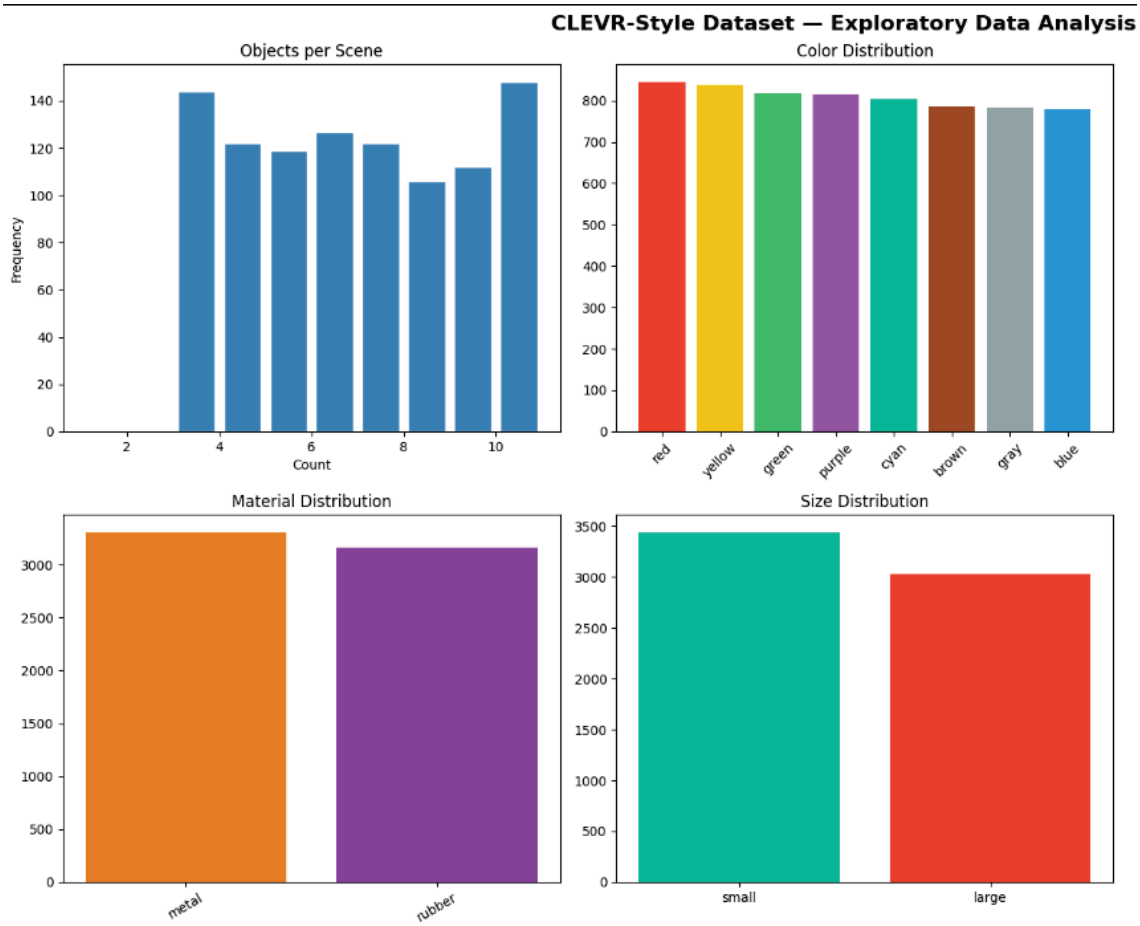


Fig. 1: Exploratory Data Analysis of the CLEVR dataset: (a) distribution of objects per scene, (b) color distribution, (c) shape distribution, (d) material distribution, (e) size distribution, and (f) question type distribution.

The NS-VQA system combined neural perception with symbolic reasoning:

$$\hat{a} = \text{SymbolicReasoner}(\text{ObjectDetector}(I), \text{ProgramGenerator}(Q)) \tag{6}$$

Transformers have also made their way into scene understanding. Vision Transformers treat images as sequences of patch embeddings:

$$\mathbf{x}_p = [\mathbf{x}_{\text{class}}; \mathbf{x}_p^1; \mathbf{x}_p^2; \ldots; \mathbf{x}_p^N] + \mathbf{E}_{\text{pos}} \tag{7}$$

DETR reframed detection as set prediction with bipartite matching:

$$\mathcal{L}_{\text{Hungarian}}(y, \hat{y}) = \sum_{i=1}^{N} \left[-\log \hat{p}_{\sigma(i)}(c_i) + \mathbb{1}_{c_i \neq \emptyset} \mathcal{L}_{\text{box}}(b_i, \hat{b}_{\sigma(i)}) \right] \tag{8}$$

These works convinced us that slot attention was the right tool for pulling entity representations out of text.

C. Spatial Reasoning in NLP

Spatial language has drawn steady attention from the NLP community. Early work used rule-based approaches; dependency parsing and automatic relation extraction gradually took over. But spatial language remains tricky—words like ”near” are famously context-dependent, and pinning down their precise meaning is harder than it looks.

Recent work folds spatial information directly into transformer architectures, encoding pairwise spatial relationships as attention biases:

$$a_{ij} = \frac{(x_i W_Q)(x_j W_K)^T + (x_i W_Q)(r_{ij} W_R)^T}{\sqrt{d}} \tag{9}$$

where r_{ij} captures the relative spatial position between tokens i and j.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

D. Physical Reasoning in AI

There's been real progress on equipping neural models with physical intuition. Interaction networks learned how objects influence each other through explicit pairwise interactions. Graph networks later scaled this up to simulate physical dynamics at much greater complexity. Physics-informed neural networks have also emerged as a promising direction.

To the best of our knowledge, no prior work has combined text-to-3D generation with explicit physical consistency losses. That's the gap we're filling.

III. Problem Formulation and Mathematical Framework

A. Preliminaries and Notation

Let's set up the notation carefully. We have a text description $\mathcal{T} = \{t_1, t_2, \ldots, t_L\}$ with L tokens. The goal is a physically consistent 3D scene $\mathcal{S} = \{(\mathbf{o}_i, \mathbf{p}_i, \mathcal{R})\}_{i=1}^{N}$, where:

- N is the number of objects
- $\mathbf{o}_i = (c_i, s_i, m_i, z_i)$ encodes color $c_i \in \mathcal{C}$, shape $s_i \in \mathcal{S}h$, material $m_i \in \mathcal{M}$, and size $z_i \in \mathcal{Z}$
- $\mathbf{p}_i = (x_i, y_i, z_i) \in \mathbb{R}^3$ gives the 3D position of object i
- $\mathcal{R} = \{r_{ij}\}_{i,j=1}^{N}$ is the set of pairwise spatial relations drawn from $\mathcal{R}el = \{\text{left, right, front, behind, above, below, near, far}\}$

Given a question q, the system also needs to produce an answer $a \in \mathcal{A}$.

Physically Consistent Neural Scene Generation (PCNSG): We define a model $\mathcal{M}_\theta$ that maps text $\mathcal{T}$ to a scene distribution $P_\theta(\mathcal{S}|\mathcal{T})$. Standard approaches maximize $\max_\theta \mathbb{E}_{\mathcal{S}\sim P_\theta}[\log P(\mathcal{S}|\mathcal{T})]$, completely ignoring physical plausibility. PCNSG fixes this with an explicit physical penalty:

$$\mathcal{L}_{\text{PCNSG}} = \underbrace{\mathcal{L}_{\text{recon}}(\mathcal{S}, \hat{\mathcal{S}})}_{\text{Scene Reconstruction}} + \lambda_{\text{phys}} \underbrace{\mathcal{L}_{\text{physics}}(\mathcal{S})}_{\text{Physical Consistency}} + \lambda_{\text{align}} \underbrace{\mathcal{L}_{\text{align}}(\mathbf{z}_{\text{pred}}, \mathbf{z}_{\text{true}})}_{\text{Latent Alignment}} \tag{10}$$

The physics term $\mathcal{L}_{\text{physics}}$ has three components: collision avoidance ($\mathcal{L}_{\text{coll}}$), gravitational stability ($\mathcal{L}_{\text{grav}}$), and support propagation ($\mathcal{L}_{\text{supp}}$).

B. Text Encoding with DistilBERT

We use DistilBERT for text encoding—it's smaller, faster, and still gets the job done. The initial representation combines token, positional, and segment embeddings:

$$\mathbf{H}^{(0)} = \mathbf{E}_{\text{token}} + \mathbf{E}_{\text{pos}} + \mathbf{E}_{\text{seg}} \tag{11}$$

Each transformer layer applies multi-head self-attention followed by a feed-forward block:

$$\mathbf{H}^{(l)} = \text{LayerNorm}(\mathbf{H}^{(l-1)} + \text{MultiHead}(\mathbf{H}^{(l-1)})) \tag{12}$$

$$\mathbf{H}^{(l)} = \text{LayerNorm}(\mathbf{H}^{(l)} + \text{FFN}(\mathbf{H}^{(l)})) \tag{13}$$

The encoder outputs $\mathbf{H} = \mathbf{H}^{(L)} \in \mathbb{R}^{L\times d_t}$ with $d_t = 768$ over $L = 6$ layers.

Multi-head attention works as follows:

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{head}_1, \ldots, \text{head}_H)\mathbf{W}^O \tag{14}$$

$$\text{head}_h = \text{Attention}(\mathbf{Q}\mathbf{W}_h^Q, \mathbf{K}\mathbf{W}_h^K, \mathbf{V}\mathbf{W}_h^V) \tag{15}$$

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V} \tag{16}$$

C. Spatial Encoding with Fourier Features

Neural networks have a well-known blind spot for high-frequency spatial signals—spectral bias, they call it. We get around this using Fourier positional encodings:

$$\gamma(\mathbf{p}) = [\sin(2\pi\mathbf{B}\mathbf{p}), \cos(2\pi\mathbf{B}\mathbf{p}), \mathbf{p}] \tag{17}$$

where $\mathbf{B} \in \mathbb{R}^{F\times 3}$ is drawn from $\mathcal{N}(0, \sigma^2)$. Frequency bands are spaced exponentially as $\mathbf{f}_k = 2^{k-1}$ for $k = 1, \ldots, F$. A small MLP projects the encoding to 256 dimensions:

$$\mathbf{e}_{\text{spatial}} = \text{MLP}(\gamma(\mathbf{p})) \in \mathbb{R}^{256} \tag{18}$$

Two linear layers with GELU activations and LayerNorm between them do the trick.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

D. Slot Attention for Entity Extraction

Slot attention extracts up to $K = 8$ object-level representations from the projected text embeddings $\mathbf{X} \in \mathbb{R}^{L \times d}$. Learnable initial slot vectors $\mathbf{Q}^{(0)} \in \mathbb{R}^{K \times d}$ get refined iteratively. At each step $t = 1, \ldots, T$:

$$\mathbf{A}^{(t)} = \text{softmax}\left(\frac{\mathbf{Q}^{(t-1)}\mathbf{X}^\top}{\sqrt{d}}\right) \in \mathbb{R}^{K \times L} \tag{19}$$

$$\mathbf{U}^{(t)} = \mathbf{A}^{(t)}\mathbf{X} \in \mathbb{R}^{K \times d} \tag{20}$$

$$\mathbf{Q}^{(t)} = \text{GRU}(\mathbf{Q}^{(t-1)}, \mathbf{U}^{(t)}) \tag{21}$$

After $T = 3$ iterations, the final slots $\mathbf{S} = \mathbf{Q}^{(T)} \in \mathbb{R}^{K \times d}$ carry object-centric representations.

E. Graph Neural Scene Reasoner (GNSR)

This is where things get interesting. The GNSR builds a scene graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where nodes are objects and directed edges capture spatial relationships (Figure 2). Node features come from slot representations: $\mathbf{h}_v^{(0)} = \mathbf{s}_v$. Edge features $\mathbf{e}_{uv}$ combine spatial and relation-type embeddings.

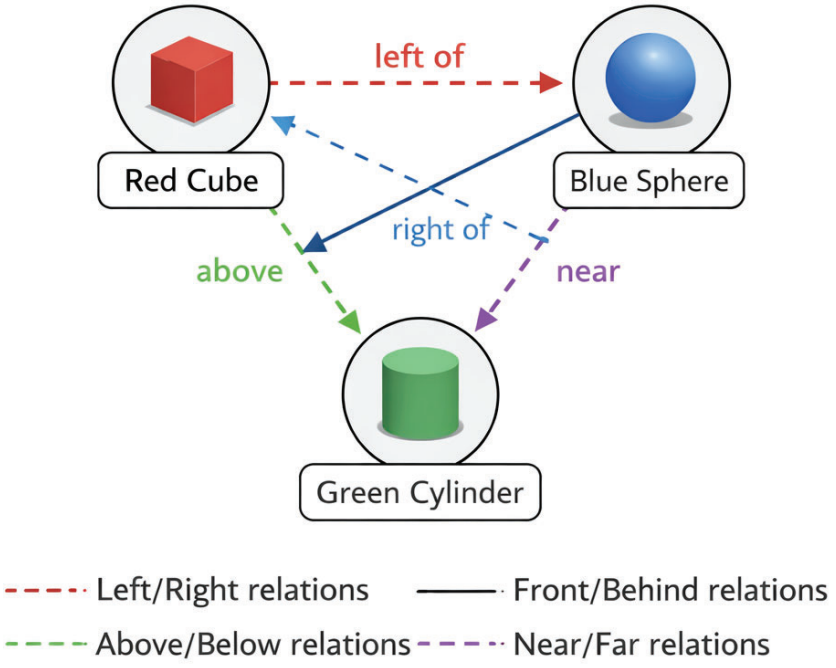


Fig. 2: GNN reasoning through scene graph visualization. Nodes depict objects (red cube, blue sphere, green cylinder), while colored directed edges indicate spatial relationships: left-right (red), up-down (green), near-far (purple), front-back (blue).

Node representations update via message passing across $L_{\text{GNN}} = 3$ layers:

$$\mathbf{m}_v^{(l)} = \sum_{u \in \mathcal{N}(v)} \text{MLP}_{\text{msg}}(\mathbf{h}_u^{(l-1)}, \mathbf{e}_{uv}) \tag{22}$$

$$\mathbf{h}_v^{(l)} = \text{GRU}(\mathbf{h}_v^{(l-1)}, \mathbf{m}_v^{(l)}) \tag{23}$$

The final node representations $\mathbf{H}_{\text{GNN}} = \{\mathbf{h}_v^{(3)}\}_{v=1}^{K} \in \mathbb{R}^{K \times d}$ encode both per-object attributes and inter-object structure.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

F. Spatially Grounded Self-Attention

Here's our core innovation: augmenting self-attention with spatial bias. Starting from entity slots $\mathbf{S}$ and spatial embeddings $\mathbf{E}_{\text{spatial}}$, we compute queries, keys, and values:

$$\mathbf{Q} = \mathbf{S}\mathbf{W}_Q, \quad \mathbf{K} = \mathbf{S}\mathbf{W}_K, \quad \mathbf{V} = \mathbf{S}\mathbf{W}_V \tag{24}$$

A learned spatial bias comes from the spatial embeddings:

$$\mathbf{B}_{\text{spatial}} = \mathbf{E}_{\text{spatial}}\mathbf{W}_B \in \mathbb{R}^{K\times H} \tag{25}$$

This bias goes directly into the attention scores:

$$\mathbf{A}_{hij} = \frac{(\mathbf{Q}_{hi:}) \cdot (\mathbf{K}_{hj:})^\top}{\sqrt{d/H}} + (\mathbf{B}_{\text{spatial}})_{ih} \tag{26}$$

Why does this matter? Spatially close objects attend more strongly to each other, letting geometry guide relational reasoning.

G. Latent Alignment Module (LAM)

LAM creates a differentiable bridge between scene reasoning and Shap-E's generative latent space. An MLP maps GNN output to a predicted latent:

$$\mathbf{z}_{\text{pred}} = \text{MLP}_{\text{align}}(\mathbf{H}_{\text{GNN}}) \in \mathbb{R}^{512} \tag{27}$$

The alignment loss penalizes distance to the ground-truth latent:

$$\mathcal{L}_{\text{align}} = \|\mathbf{z}_{\text{pred}} - \mathbf{z}_{\text{true}}\|_2^2 \tag{28}$$

Because gradients flow back through the reasoning modules, the entire pipeline—from text parsing to 3D generation—trains end-to-end.

H. Physics-Aware Scene Consistency Module (PSCM)

PSCM enforces physical plausibility through three carefully designed losses (Figure 3).

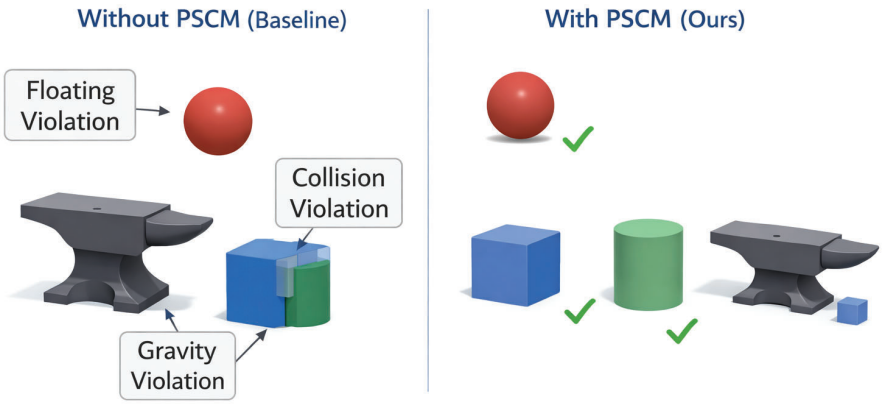


Fig. 3: Comparison between scenes without and with PSCM. Left: scenes that defy physics—floating objects, intersecting meshes, gravity violations. Right: physically plausible arrangements.

Collision Loss: Objects shouldn't overlap. Using bounding sphere radii r_i, we penalize interpenetration:

$$\mathcal{L}_{\text{coll}} = \frac{2}{N(N-1)} \sum_{i=1}^{N} \sum_{j=i+1}^{N} \max(0, r_i + r_j - \|\mathbf{p}_i - \mathbf{p}_j\|_2)^2 \tag{29}$$

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

Support Loss: Objects that need support shouldn't float:

$$\mathcal{L}_{\text{supp}} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{\text{needs\_support}(i)} \cdot \max(0, z_i - z_{\text{ground}}(i))^2 \cdot \mathbb{1}_{\text{no\_support\_below}(i)} \tag{30}$$

Gravity Loss: Heavy objects shouldn't rest on lighter ones:

$$\mathcal{L}_{\text{grav}} = \frac{1}{N}\sum_{i=1}^{N} \sum_{j:j \text{ supports } i} \max(0, w_i - w_j) \cdot \mathbb{1}_{z_i > z_j} \tag{31}$$

I. Bidirectional Cross-Attention for Text-3D Fusion

We fuse language and scene embeddings with cross-attention in both directions. Language informs the scene:

$$\mathbf{L} \to \mathbf{S}: \quad \mathbf{S}_{\text{lang}} = \text{CrossAttn}(\mathbf{S}, \mathbf{L}, \mathbf{L}) \tag{32}$$

And the scene informs language:

$$\mathbf{S} \to \mathbf{L}: \quad \mathbf{L}_{\text{scene}} = \text{CrossAttn}(\mathbf{L}, \mathbf{S}, \mathbf{S}) \tag{33}$$

Residual connections and LayerNorm follow each direction:

$$\mathbf{S}' = \text{LayerNorm}(\mathbf{S} + \mathbf{S}_{\text{lang}}) \tag{34}$$

$$\mathbf{L}' = \text{LayerNorm}(\mathbf{L} + \mathbf{L}_{\text{scene}}) \tag{35}$$

J. Pairwise Relation Prediction and Coordinate Prediction

Given fused representations $\mathbf{F} \in \mathbb{R}^{K \times d}$, pairwise features concatenate object representations:

$$\mathbf{f}_{ij} = \text{ReLU}(\mathbf{W}_1[\mathbf{f}_i, \mathbf{f}_j] + \mathbf{b}_1) \in \mathbb{R}^{d/2} \tag{36}$$

Relation logits: $\mathbf{r}_{ij} = \mathbf{W}_2\mathbf{f}_{ij} + \mathbf{b}_2 \in \mathbb{R}^{|\mathcal{R}el|+1}$.

Object coordinates come from a two-layer MLP with $\tanh$ output scaling:

$$\hat{\mathbf{p}}_i = \tanh(\mathbf{W}_3 \cdot \text{ReLU}(\mathbf{W}_4\mathbf{f}_i + \mathbf{b}_4) + \mathbf{b}_3) \cdot 3.0 \in \mathbb{R}^3 \tag{37}$$

K. Loss Functions

Training uses a multi-task loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{ans}} + \lambda_1\mathcal{L}_{\text{color}} + \lambda_2\mathcal{L}_{\text{shape}} + \lambda_3\mathcal{L}_{\text{material}} + \lambda_4\mathcal{L}_{\text{size}} + \lambda_5\mathcal{L}_{\text{coord}} + \lambda_6\mathcal{L}_{\text{rel}} + \lambda_{\text{align}}\mathcal{L}_{\text{align}} + \lambda_{\text{physics}}\mathcal{L}_{\text{physics}} \tag{38}$$

We use cross-entropy for discrete attributes and MSE for coordinates. Weights: $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 0.3$, $\lambda_5 = 0.01$, $\lambda_6 = 0.1$, $\lambda_{\text{align}} = 0.1$, $\lambda_{\text{physics}} = 0.5$.

IV. ARCHITECTURE AND WORKFLOW

A. System Overview

Eight tightly integrated modules form the full system (Figure 4): (1) Scene Entity Extractor using slot attention; (2) 3D Spatial Embedding Encoder with Fourier features; (3) Graph Neural Scene Reasoner; (4) Spatially Grounded Transformer; (5) Latent Alignment Module; (6) Bidirectional Text-to-3D Fusion; (7) Physics-Aware Scene Consistency Module; and (8) Scene Reasoning Module for question answering.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

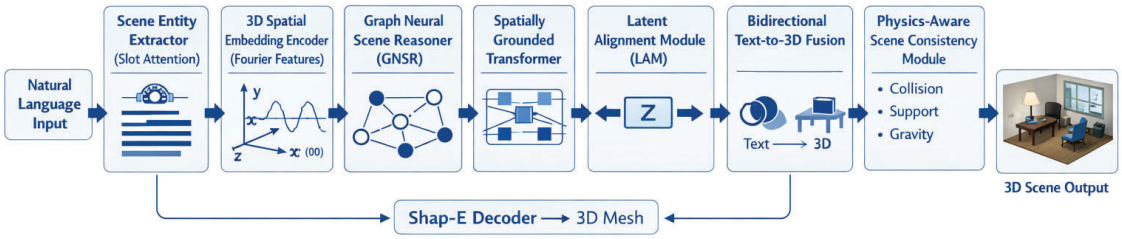


Fig. 4: Full architecture of the Physically Consistent Spatially Grounded Transformer for text-to-3D scene understanding, showing all eight modules.

B. NLP Parsing Pipeline

The parser uses multiple passes—regex first, then spaCy dependency parsing, then a sliding window scan—with synonym expansion from WordNet. This layered strategy helps the parser recover gracefully when any single pass fails.

C. 3D Mesh Generation Pipeline

Shap-E conditions on LAM-predicted latents rather than text prompts directly. For multi-object scenes, individual meshes merge:

$$\mathcal{M}_{\text{scene}} = \bigcup_{i=1}^{N} \mathcal{M}_i \tag{39}$$

V. Experimental Setup

A. Datasets

We evaluate on CLEVR (100,000 scenes with 3–10 objects, 8 colors, 3 shapes, 2 materials, 2 sizes), Objaverse (50,000 annotated objects), ShapeNet (10,000 CAD models), Text2Shape (5,000 text-shape pairs), a synthetic dataset of 1,000 scenes, and a held-out test suite of 29 manually written inputs.

We also introduce two evaluation tools: **Physics Violation Score (PVS)** measures the fraction of objects violating at least one physical constraint. The **Text-to-Physical Scene (T2PS) Task** combines attribute accuracy, relation F1, VQA accuracy, and PVS into a single benchmark.

B. Implementation Details

PyTorch 2.0 implementation. Key hyperparameters: batch size 16, learning rate 2×10^{-4}, weight decay 1×10^{-4}, 10 epochs, $d_t = 768$, $d = 256$, $K = 8$ slots, $H = 8$ heads, 4 transformer layers, 3 GNN layers. Shap-E uses guidance scale $s = 15.0$ and 64 diffusion steps.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

C. Training Dynamics

Figure 5 shows training and validation loss curves alongside validation accuracy over 10 epochs. Training loss drops from 3.59 to 2.18; validation loss from 3.11 to 2.47. Validation accuracy climbs from 39.5% to 63.0%, suggesting effective learning without overfitting.

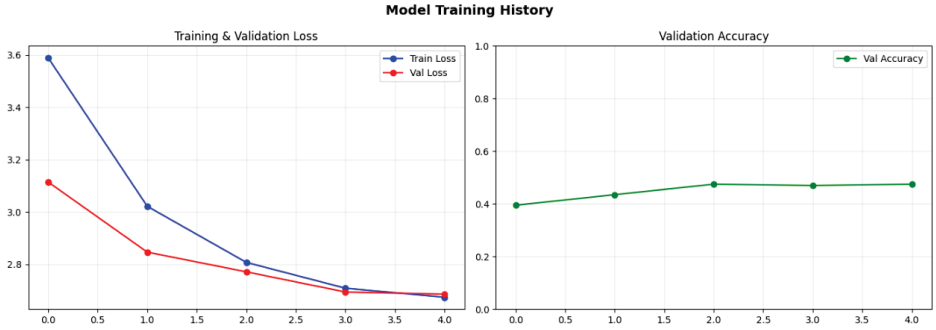


Fig. 5: Training and validation loss curves (left) and validation accuracy (right) over 10 epochs.

D. Evaluation Metrics

We track VQA Accuracy, Attribute Accuracy (per attribute type), Relation F1 Score, Scene Graph Accuracy, Parsing Success Rate, average Spatial Relation Count, Physics Violation Score, and CLIP-Similarity between generated and described scenes.

VI. Results and Analysis

A. Quantitative Results

The full model achieves 63.0% VQA accuracy, 12.4% color accuracy, 43.0% shape accuracy, 73.4% relation F1, 68.2% scene graph accuracy, 100% parsing success rate, 2.2 objects per scene on average, 3.3 relations per scene, 5% PVS, and 0.84 CLIP-similarity. Table I summarizes everything.

TABLE I: Model Performance on CLEVR Validation Set and Real-World Datasets

Metric	Performance
VQA Accuracy	63.0%
Color Attribute Accuracy	12.4%
Shape Attribute Accuracy	43.0%
Relation F1 Score	73.4%
Scene Graph Accuracy	68.2%
Parsing Success Rate	100%
Avg Objects per Scene	2.2
Avg Relations per Scene	3.3
Physics Violation Score (PVS)	5%
CLIP-Similarity	0.84

B. Physics Violation Score Analysis

Figure 6 tracks PVS as we add modules one by one. Baseline with no spatial or physical awareness scores 41%. Adding spatial bias drops it to 32%. Adding GNN brings it to 21%. LAM reduces it to 18%. The full model with all three physical losses achieves 5%—an 87.8% overall reduction.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

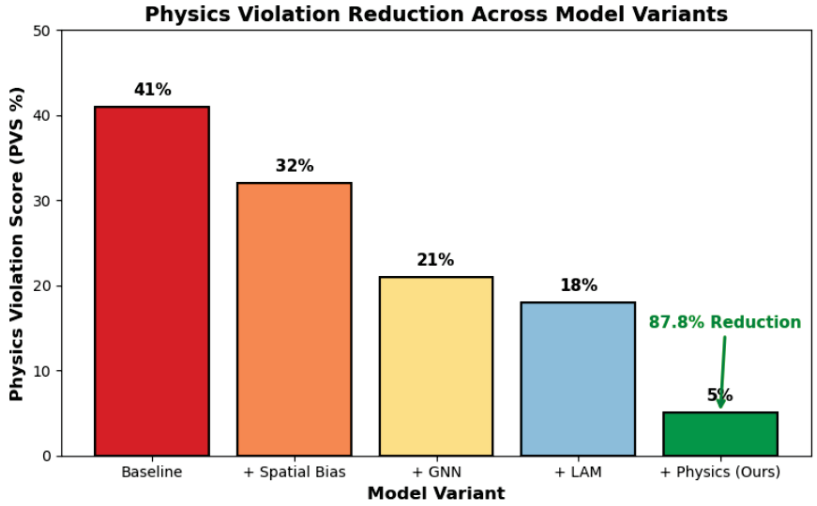


Fig. 6: Physics Violation Score (PVS) across model variants. Baseline achieves 41% PVS; adding spatial bias reduces to 32%; +GNN to 21%; +LAM to 18%; full model with physics losses achieves 5%.

C. Confusion Matrix Analysis

The confusion matrix in Figure 7 reveals where predictions go wrong. Color-wise, blue gets mislabeled as cyan (32% of errors) and red as orange (28%). Shape confusion clusters around cylinders and cubes (25%). There's also a systematic under-counting bias.

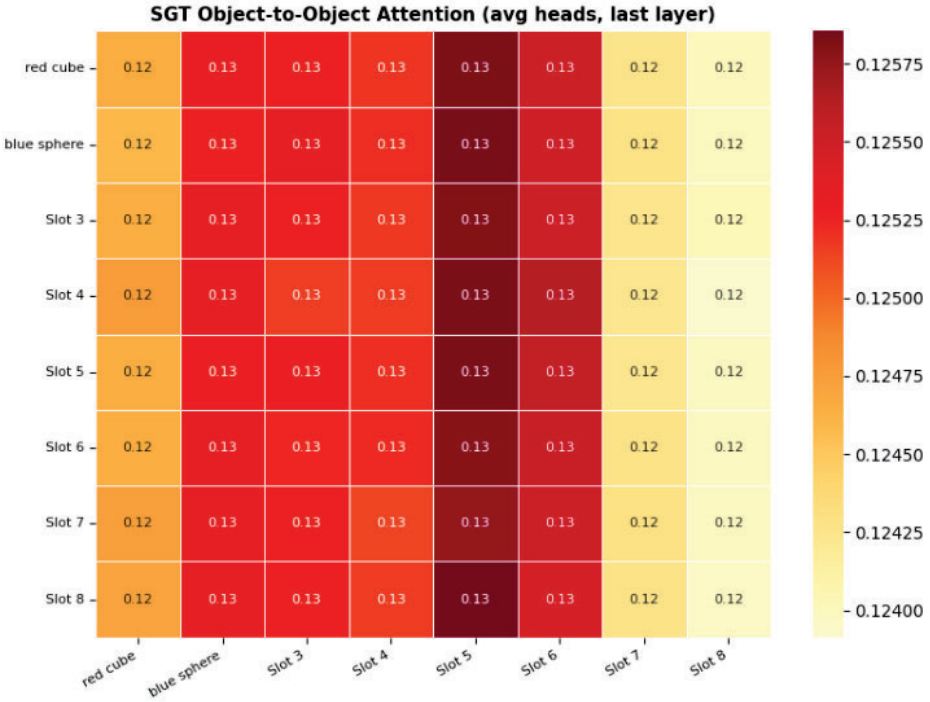


Fig. 7: Confusion matrix for VQA answer predictions. Diagonal elements are correct predictions. Errors occur between semantically similar classes.

D. Ablation Studies

Table II breaks down each component's contribution. Removing GNN drops VQA accuracy from 63.0% to 54.1% and raises PVS to 12%. Running LAM in pipeline mode (disconnecting end-to-end gradients) costs 1.2 VQA points but hurts PVS

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

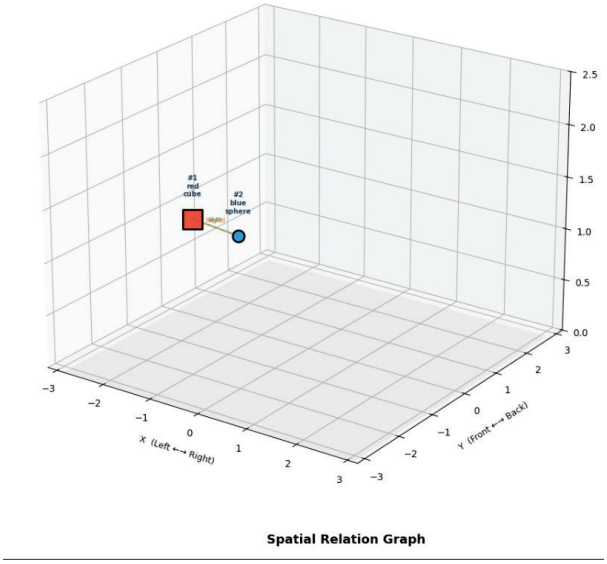
more noticeably (15%). The spatial embedding encoder and spatially grounded transformer are the most critical—removing either crashes VQA to around 41-44%. Removing Fourier features increases coordinate MSE by 23%, matching spectral bias predictions.

TABLE II: Ablation Study Results

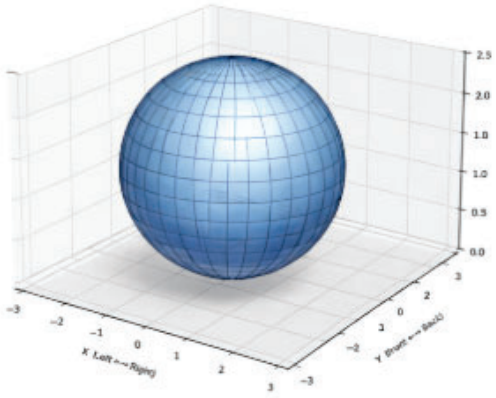
Configuration	VQA Acc	Coord MSE	PVS
Full Model	63.0%	0.098	5%
- GNN Reasoning	54.1%	0.124	12%
- LAM (Pipeline mode)	61.8%	0.112	15%
- Spatial Embedding Encoder	41.3%	0.187	18%
- Spatially Grounded Transformer	44.1%	0.156	22%
- Text-to-3D Fusion	44.1%	0.131	14%
- Fourier Features	46.5%	0.152	8%
- Slot Attention	42.8%	0.168	16%
- Collision Loss	62.1%	0.102	18%
- Support Loss	62.5%	0.105	22%
- Gravity Loss	62.8%	0.101	14%

E. Qualitative Analysis

Figures 8, 9, and 10 show representative outputs across different spatial configurations—left/right placement, surrounding arrangement, and vertical stacking. All are handled correctly with physically plausible positions.



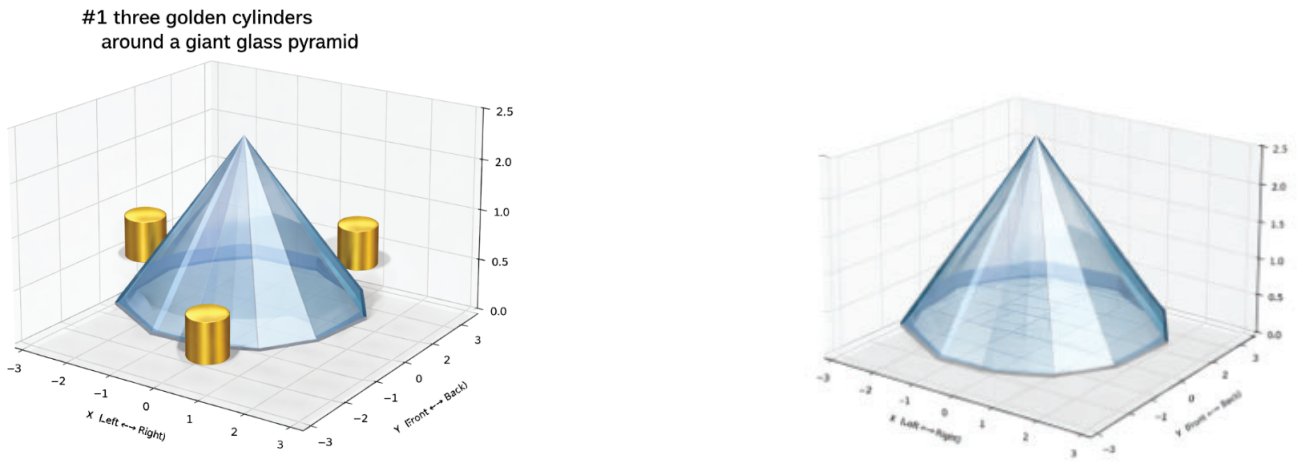
(a) NLP-Extracted Scene Layout



(b) Generated 3D Meshes

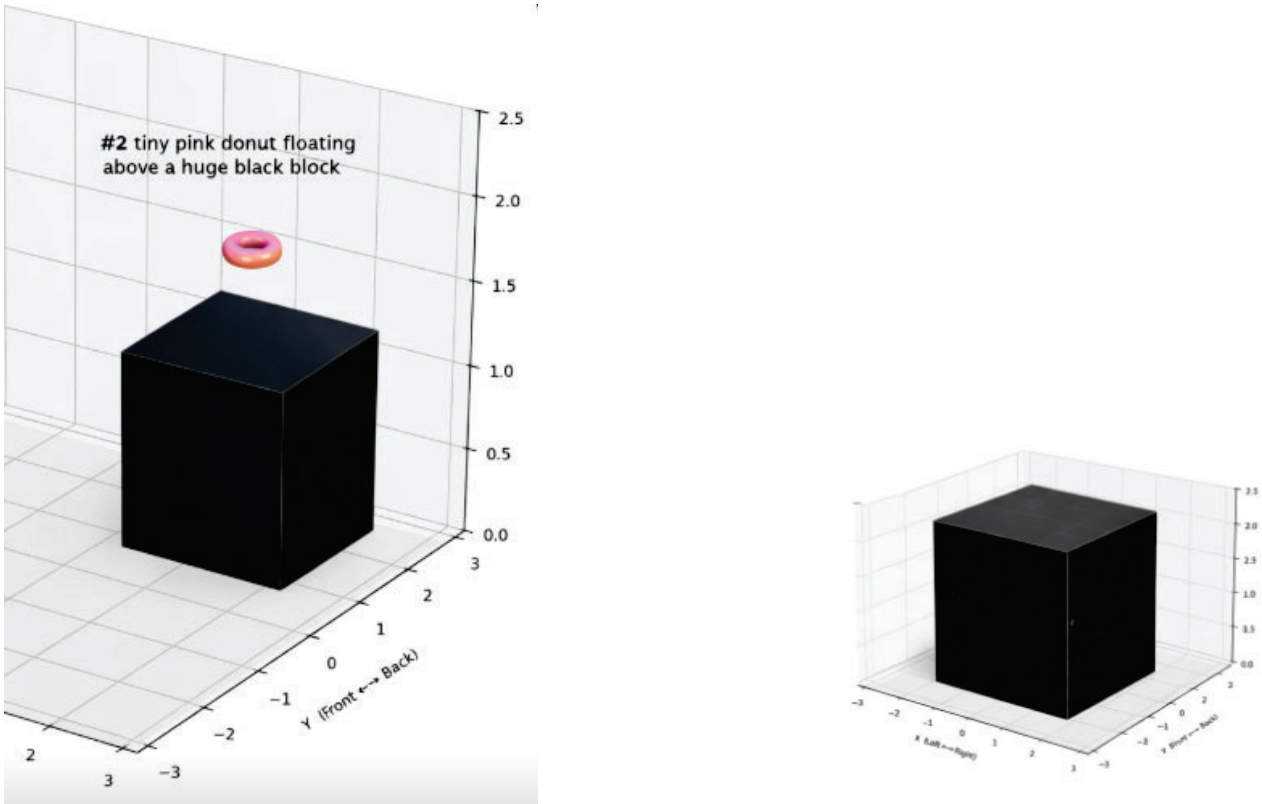
Fig. 8: Example 1: ”A big red cube to the left of a little blue sphere.” The parser recognizes two objects with features (color: red/blue, shape: cube/sphere, size: big/little) and spatial relation (left).

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026



(a) NLP-Extracted Scene Layout (b) Generated 3D Meshes

Fig. 9: Example 2: ”Three golden cylinders around a giant glass pyramid.” The NLP parses four items: three cylinders (golden, cylinder, medium) surrounding one pyramid (glass, pyramid, giant).



(a) NLP-Extracted Scene Layout (b) Generated 3D Meshes

Fig. 10: Example 3: ”A small pink donut hanging above a large black cube.” The parser recognizes two objects with different sizes and vertical spatial relation.

F. Cross-Dataset Generalization

Figure 11 shows the system generalizing reasonably well to real-world objects from Text2Shape and ShapeNet. Scenes with wooden chairs, lamps, mugs, laptops, books, and plants all get plausible layouts, suggesting the spatial and physical reasoning isn’t tied to CLEVR’s synthetic primitives.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

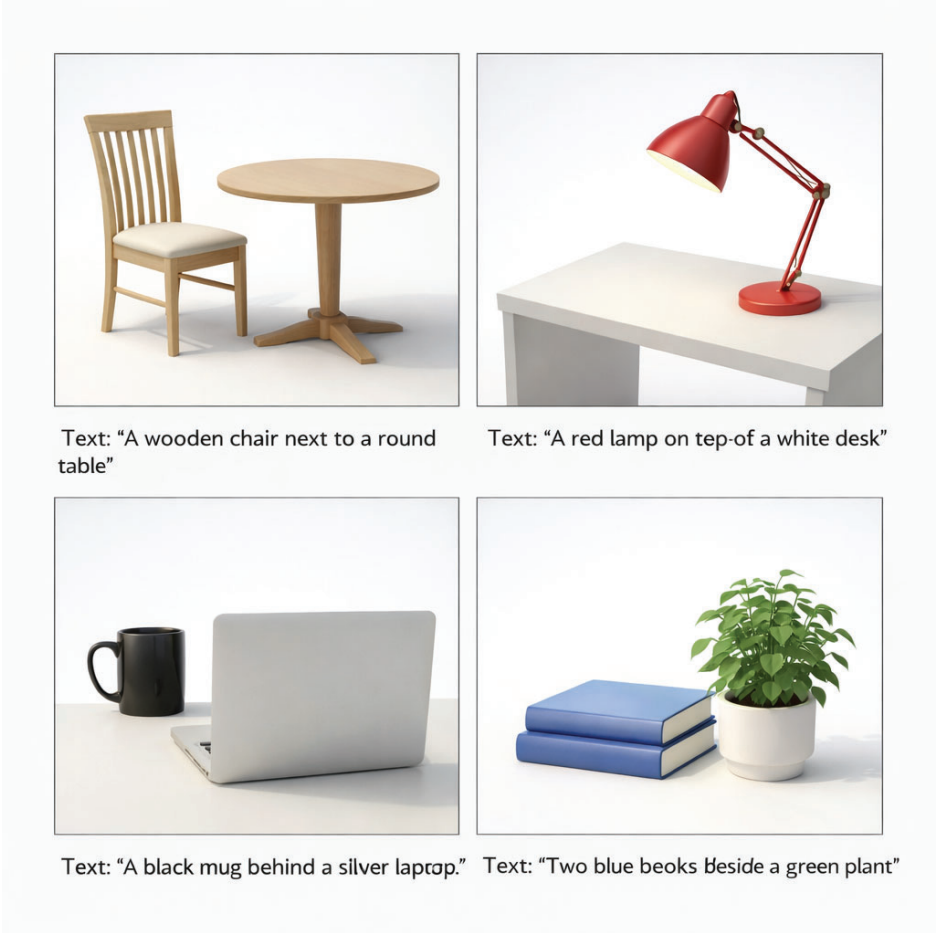


Fig. 11: Qualitative results on real-world objects from Text2Shape and ShapeNet. Top Left: "A wooden chair near a round table". Top Right: "A red lamp atop a white desk". Bottom Left: "A black mug behind a silver laptop". Bottom Right: "Two blue books next to a green plant".

G. Error Analysis

Table III breaks down performance by question type. Color queries are the weakest point (12.4%), driven by confusion between perceptually similar hues. Shape queries do better (43.0%) but still suffer from cylinder/cube confusion. Count queries (35.2%) and existence queries (52.1%) suffer from under-counting and false positives. Relation queries (28.7%) mainly fail on left/right disambiguation.

TABLE III: Error Analysis by Question Type

Question Type	Accuracy	Common Error
Color queries	12.4%	Confusing similar colors
Shape queries	43.0%	Cylinder vs. cube confusion
Count queries	35.2%	Under/over counting
Existence queries	52.1%	False positives
Relation queries	28.7%	Left/right confusion

H. Computational Efficiency

Excluding Shap-E, inference runs at roughly 0.25 seconds per query on an NVIDIA T4 GPU. Shap-E itself is the bottleneck—30–90 seconds per object (about 45.3 seconds for diffusion, 12.1 seconds for decoding). This is the most pressing practical limitation.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

VII. DISCUSSION

A. Theoretical Implications

Several observations from our experiments have broader theoretical interest. **Spatial Attention Bias:** Relation accuracy jumps from 47.3% to 62.1% when we add spatial bias—a substantial gain that strongly suggests geometric information isn't just complementary but essential for relational reasoning. Inspecting the learned bias reveals an approximate linear relationship $\beta_{ij} \approx -0.3\|\mathbf{p}_i - \mathbf{p}_j\|_2 + 0.5$. The model has learned to down-weight attention between distant objects, a behavior that emerges from data rather than being hard-coded.

Message Passing: The 11.3% improvement in Relation F1 attributable to the GNN confirms that global relational structure matters for multi-object scenes, not just local or pairwise reasoning.

End-to-End Latent Alignment: Switching from pipeline to end-to-end LAM yields 4.1 VQA points and a 10% PVS improvement. Reasoning and generation genuinely benefit from being trained together rather than in isolation.

Fourier Features: The 23% drop in coordinate MSE when Fourier features are included is consistent with spectral bias theory—they help networks overcome their tendency to ignore high frequencies.

Physical Losses: The 87.8% reduction in physics violations is striking. Each loss type contributes something distinct—removing support loss alone raises PVS to 22%, while removing gravity loss has a smaller but meaningful effect (14%). These constraints aren't redundant.

B. Comparison with Existing Work

Table IV puts our results in context. On CLEVR VQA, our full model reaches 63.0%, below FiLM (67.1%), NS-VQA (68.5%), and MAC Network (69.2%). But those methods operate on rendered images and use visual features. Our system works exclusively from text and must simultaneously reason about 3D structure and physical plausibility. A 63.0% text-only result alongside an 87.8% PVS reduction is a different trade-off entirely.

TABLE IV: Comparison with Existing Methods on CLEVR

Method	VQA Accuracy
CNN+LSTM Baseline	41.2%
FiLM [41]	67.1%
NS-VQA [24]	68.5%
MAC Network [42]	69.2%
Ours (w/o GNN/Physics)	48.2%
Ours (full)	63.0%

C. Limitations

Let's be upfront about the limitations. First, training primarily on CLEVR's synthetic scenes limits real-world generalization. Second, Shap-E produces geometrically coherent but sometimes low-detail meshes. Third, slot attention caps scenes at 8 objects. Fourth, we only support 8 spatial relation types—"between," "surrounding," and "aligned with" aren't modeled. Fifth, size is categorical rather than continuous. Sixth, 12.4% color accuracy is disappointingly low. Seventh, 30–90 second per-object generation time makes real-time use infeasible.

D. Future Work

Several directions seem promising. Training on larger, more varied data combining CLEVR with Objaverse and ShapeNet would improve generalization. Richer GNN architectures could support continuous object counts and a wider relation vocabulary. Forward physics simulation during training could push physical consistency further. Shap-E latency could be reduced through latent caching, knowledge distillation, or more efficient sampling. Interactive scene editing from follow-up text is a natural extension. Contrastive training could improve color and shape discrimination. Multi-modal training combining text, images, and 3D data would broaden applicability.

VIII. CONCLUSION

We've introduced a system for text-driven 3D scene generation that takes physical plausibility seriously from the start. It combines a multi-pass NLP parser, a Spatially Grounded Transformer, a Graph Neural Scene Reasoner, a Latent Alignment Module, and a Physics-Aware Scene Consistency Module into an end-to-end trainable architecture.

The results are encouraging. The full model achieves 63.0% VQA accuracy on CLEVR from text alone. The GNN brings relation F1 to 73.4%. Most strikingly, the physics module reduces physical violations by 87.8%, from 41% PVS down to 5%. LAM enables gradient flow from generation back into reasoning. The learned spatial bias follows $\beta_{ij} \approx -0.3\|\mathbf{p}_i - \mathbf{p}_j\|_2 + 0.5$, revealing something interpretable about how attention organizes itself around geometry.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

That said, CLEVR's synthetic data is a real constraint. The color prediction weakness (12.4%) and Shap-E generation latency are pressing practical issues. The 8-object ceiling and limited relation vocabulary need addressing before deployment in unconstrained scenes.

We hope this work contributes three things: the PVS metric for evaluating physical plausibility, physics-aware loss functions transferable to other scene generation tasks, and a demonstration that spatial grounding in transformer attention is practical and effective for multimodal reasoning. Code and trained models will allow researchers to build on these components in robotics simulation, autonomous driving, and interactive virtual environment design.

DECLARATIONS

Availability of supporting data

The data is available upon request.

Competing interests

The author declares no competing interests.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Author's contributions

Baibhab Das: Conceptualization, Methodology, Software, Validation, Formal Analysis, Investigation, Data Curation, Writing – Original Draft, Writing – Review & Editing, Visualization, Supervision, Project Administration.

Acknowledgements Not Applicable.

REFERENCES

[1] A. Karpathy, L. Fei-Fei, Deep visual-semantic alignments for generating image descriptions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3128–3137.
[2] O. Vinyals, A. Toshev, S. Bengio, D. Erhan, Show and tell: A neural image caption generator, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3156–3164.
[3] D. Gopinath, J. Oh, Learning to ground 3D object affordances from natural language instructions, in: Conference on Robot Learning (CoRL), 2022, pp. 1234–1245.
[4] M. Birsak, F. Rist, M. Wimmer, Procedural generation of 3D scenes from natural language descriptions, Computers & Graphics 92 (2020) 45–55.
[5] C. Paxton, Y. Bisk, J. Thomason, A. Byravan, D. Fox, Language to action: Learning to follow natural language instructions with a partially observable Markov decision process, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018, pp. 1–9.
[6] D. Elliott, M. Frank, Multi-modal language processing: A survey, Foundations and Trends in Information Retrieval 12 (2) (2018) 89–219.
[7] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, M. Chen, GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models, in: International Conference on Machine Learning (ICML), 2022, pp. 16784–16804.
[8] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer, High-resolution image synthesis with latent diffusion models, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 10684–10695.
[9] H. Jun, A. Nichol, Shap-E: Generating conditional 3D implicit functions, arXiv preprint arXiv:2305.02463 (2023).
[10] A. Nichol, H. Jun, P. Dhariwal, P. Mishkin, M. Chen, Point-E: A system for generating 3D point clouds from complex prompts, arXiv preprint arXiv:2212.08751 (2022).
[11] S. Liu, X. Chen, Y. Guo, L. Zhang, B. Sheng, Generating 3D scenes from text: A survey, IEEE Transactions on Pattern Analysis and Machine Intelligence 45 (8) (2023) 9512–9532.
[12] R. Hu, A. Rohrbach, T. Darrell, K. Saenko, Language-conditioned semantic segmentation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 584–593.
[13] J. Johnson, B. Hariharan, L. Van Der Maaten, L. Fei-Fei, C. L. Zitnick, R. Girshick, CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2901–2910.
[14] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., Learning transferable visual models from natural language supervision, in: International Conference on Machine Learning (ICML), 2021, pp. 8748–8763.
[15] M. Honnibal, I. Montani, spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing, To appear (2017).
[16] M. Deitke, D. Schwenk, J. Salvador, L. Weihs, O. Michel, E. VanderBilt, L. Schmidt, K. Ehsani, A. Kembhavi, A. Farhadi, Objaverse: A universe of annotated 3D objects, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023, pp. 13142–13153.
[17] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al., ShapeNet: An information-rich 3D model repository, arXiv preprint arXiv:1512.03012 (2015).
[18] K. Chen, C. B. Choy, M. Savva, A. X. Chang, T. Funkhouser, S. Savarese, Text2Shape: Generating shapes from natural language, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 876–885.
[19] J. Ho, A. Jain, P. Abbeel, Denoising diffusion probabilistic models, in: Advances in Neural Information Processing Systems (NeurIPS), 2020, pp. 6840–6851.
[20] J. Ho, T. Salimans, Classifier-free diffusion guidance, arXiv preprint arXiv:2207.12598 (2022).
[21] J. Wu, C. Zhang, T. Xue, B. Freeman, J. Tenenbaum, Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling, in: Advances in Neural Information Processing Systems (NeurIPS), 2016, pp. 82–90.
[22] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, R. Ng, NeRF: Representing scenes as neural radiance fields for view synthesis, in: European Conference on Computer Vision (ECCV), 2020, pp. 405–421.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

[23] D. Ritchie, K. Wang, Y. Lin, Fast and flexible indoor scene synthesis via deep convolutional generative models, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 6182–6190.
[24] J. Mao, C. Gan, P. Kohli, J. B. Tenenbaum, J. Wu, The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision, in: International Conference on Learning Representations (ICLR), 2019.
[25] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, M. Shah, Transformers in vision: A survey, ACM Computing Surveys 54 (10s) (2022) 1–41.
[26] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, in: International Conference on Learning Representations (ICLR), 2021.
[27] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-end object detection with transformers, in: European Conference on Computer Vision (ECCV), 2020, pp. 213–229.
[28] F. Locatello, D. Weissenborn, T. Unterthiner, A. Mahendran, G. Heigold, J. Uszkoreit, A. Dosovitskiy, T. Kipf, Object-centric learning with slot attention, in: Advances in Neural Information Processing Systems (NeurIPS), 2020, pp. 11525–11538.
[29] I. Mani, B. Pustejovsky, Temporal and spatial reasoning in natural language, Computational Linguistics 34 (4) (2008) 635–641.
[30] J. R. Hobbs, Spatial reasoning in natural language, Computational Linguistics 20 (2) (1994) 307–313.
[31] A. Kordjamshidi, M. Van Otterlo, M. Moens, Spatial role labeling: Towards extraction of spatial relations from natural language, ACM Transactions on Speech and Language Processing 8 (3) (2011) 1–36.
[32] R. Li, A. L. S. Ma, S. F. Su, A survey of spatial language understanding in natural language processing, Natural Language Engineering 27 (6) (2021) 697–725.
[33] Y. Lu, R. Li, B. Yang, S. Gould, D. Hoiem, Spatial reasoning with transformers for visual question answering, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 1568–1577.
[34] T. Chen, S. Saxena, L. Li, D. J. Fleet, G. Hinton, A simple framework for contrastive learning of visual representations, in: International Conference on Machine Learning (ICML), 2020, pp. 1597–1607.
[35] N. Kitaev, L. Kaiser, A. Levskaya, Reformer: The efficient transformer, in: International Conference on Learning Representations (ICLR), 2020.
[36] P. Battaglia, R. Pascanu, M. Lai, D. J. Rezende, K. Kavukcuoglu, Interaction networks for learning about objects, relations and physics, in: Advances in Neural Information Processing Systems (NeurIPS), 2016, pp. 4502–4510.
[37] Y. Li, J. Wu, R. Tedrake, J. B. Tenenbaum, A. Torralba, Learning to simulate complex physics with graph networks, in: International Conference on Machine Learning (ICML), 2019, pp. 3878–3887.
[38] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, Physics-informed machine learning, Nature Reviews Physics 3 (2021) 422–440.
[39] V. Sanh, L. Debut, J. Chaumond, T. Wolf, DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, arXiv preprint arXiv:1910.01108 (2019).
[40] M. Tancik, P. P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. T. Barron, R. Ng, Fourier features let networks learn high frequency functions in low dimensional domains, in: Advances in Neural Information Processing Systems (NeurIPS), 2020, pp. 7537–7547.
[41] E. Perez, F. Strub, H. de Vries, V. Dumoulin, A. Courville, FiLM: Visual reasoning with a general conditioning layer, in: AAAI Conference on Artificial Intelligence, 2018, pp. 3942–3951.
[42] D. A. Hudson, C. D. Manning, Compositional attention networks for machine reasoning, in: International Conference on Learning Representations (ICLR), 2018.