NSGR: Neuro-Symbolic Grounding in Gaussian Representations

Sacchin Sundar, Anandi Arora, and Michael Tanner
{sacchin, anandiar, tannermi}@umich.edu

***Abstract*—In order to enhance 3D language grounding, this project introduces NSGR, an extension of the NS3D framework that incorporates Gaussian Splat representations. Viewpoint-dependent and multi-object relationships can be better understood thanks to NSGR's ability to capture continuous geometry and rich colour cues by substituting Gaussian-based modelling for point-cloud encoders. According to our findings, NSGR performs similarly to NS3D on indoor data, but it performs better in outdoor settings like KITTI. These results demonstrate the potential of Gaussian splats as a unified representation for neuro-symbolic 3D reasoning that is both scalable and comprehensible.**

***Index Terms*—Neuro-symbolic reasoning, 3D visual grounding, Gaussian splatting, Scene representations, Vision-language models, Outdoor scene understanding, Autonomous driving, Referring expression grounding, Data-efficient learning, Multimodal perception.**

I. Introduction

Understanding and grounding object properties and their relationships within 3D environments is a key requirement for many AI applications, including autonomous driving, embodied planning, and visually grounded dialogue. Yet, the inherent variability of 3D data introduces two core challenges: labeling is costly, and language referring to 3D scenes is complex. Therefore, effective models must be data-efficient, capable of generalizing, and able to interpret intricate language semantics such as viewpoint-dependent references and multi-object relationships.

Recent work such as NS3D [1] has shown that neuro-symbolic decompositions can provide interpretable and data-efficient 3D grounding by converting natural language into structured programs executed over learned perception modules. However, NS3D and related approaches rely on point-cloud representations of indoor scenes (e.g., ScanNet [2]), which can be sparse, noisy, and limited in their ability to express fine-grained visual attributes.

To address these limitations, we explore the use of Gaussian Splat scene representations, which provide a more continuous and photorealistic encoding of 3D geometry and appearance. Specifically, instead of using ScanNet point clouds, we use SceneSplat [3], a Gaussian-splat-based reconstruction of the same environments. This representation enables richer modeling of low-level attributes such as color—an important cue for grounding outdoor navigation language (e.g. "red stop sign" or "white speed-limit sign"). Although the training data is indoor-only, we evaluate our approach on both indoor and outdoor scenes to measure its performance and generalization ability.

Using this representation, we propose the NSGR framework, a neuro-symbolic grounding approach that extends NS3D with modules designed to operate over Gaussian splats, thus demonstrating how neuro-symbolic reasoning can be adapted to higher-fidelity 3D outdoor scene representations.

II. Related Work

1) 3D Grounding: A number of works address grounding natural language in 3D environments using point-clouds representations. ScanRefer [4] introduced the problem of locating objects in RGB-D scans from unconstrained referring expressions. Subsequent datasets and models such as ReferIt3D [5], InstanceRefer [6], and 3DVG-Transformer [7] improved grounding accuracy through better feature fusion and transformer-based reasoning. While effective, these approaches operate directly on point clouds, which are often sparse and lack the rich apperance information required for grounding attributes such as color.

2) Neuro-Symbolic Visual Reasoning Methods: Neuro-symbolic reasoning has been explored as a way to improve compositionality and data efficiency in vision-language tasks. Neural Module Networks [8] and NS-VQA [9] demonstrated the benefits of executing structured programs derived from language. NS3D [1] extended these ideas to 3D scenes by converting natural language into symbolic programs executed over learned 3D perception modules. However, NS3D relies on low-level point-cloud features from indoor environments, limiting its ability to capture fine-grained appearance cues and generalize to more visually diverse outdoor settings.

3) 3D Scene Representations: Recent advances in radiance-field representations have introduced more expressive alternatives to point clouds. 3D Gaussian Splatting [10] models scenes with continuous Gaussian primitives, enabling high-quality appearance and geometry reconstruction. SceneSplat [3] provides Gaussian-splat reconstructions of ScanNet [2], offering dense color and shape information. Such representations are promising for grounding tasks that rely on fine-grained visual attributes, such as those encountered in outdoor autonomous driving environments.

III. Methodology

Our framework extends the neuro-symbolic grounding pipeline of NS3D [1] by replacing its PointNet++ object encoder with a Gaussian Splat encoder inspired by SceneSplat [3], enabling continuous, geometry-aware, and

photometrically-rich 3D reasoning. NSGR consists of four main components: (1) semantic parsing and program generation, (2) 3D Gaussian encoding, (3) neural program execution, and (4) end-to-end training using SceneSplat-style supervision.

A. Semantic Parsing and Program Generation

Following NS3D, we translate free-form natural language utterances into executable symbolic programs composed from a domain-specific language (DSL). This program decomposition exposes the underlying reasoning graph and allows modular, compositional interpretation of high-arity 3D relationships.

We employ a language-to-code model (e.g. Codex [11]) to parse utterances into symbolic functional programs with operators such as:

- `filter(scene(), category)`
- `relate(object_set_A, object_set_B, relation)`
- `anchor(reference, target, viewpoint)`

Each operator corresponds to a neural module that consumes Gaussian-derived object or relation embeddings. This parsing mechanism preserves the advantages demonstrated in NS3D—perfect program induction with only a few prompt examples, generalization to unseen linguistic forms, and explicit handling of high-arity 3D relations such as viewpoint anchoring and multi-object reference grounding.

B. 3D Gaussian Encoder

To replace NS3D's PointNet++ point-cloud encoder, we adopt a continuous Gaussian-based representation: each scene is represented as a set of 3D Gaussians

$$G_i = (\mu_i, \Sigma_i, c_i, \alpha_i),$$

encoding mean position, anisotropic covariance, color, and opacity. Unlike discrete point clouds, Gaussian splats preserve spatial continuity, photometric features, and multi-view consistency. This representation enables stronger grounding for expressions such as "the red sign", "the tall pole behind the bench", and "the box closest to the car".

1) Object-Level Encodings: We group primitive Gaussians into object proposals (using instance labels or clustering on Gaussian positions). For each object $O_k = \{G_i\}$, we process the constituent Gaussians with a transformer encoder adapted from SceneSplat:

$$f_k^{obj} = E_{\text{GS}}(O_k),$$

where E_{GS} consumes the Gaussian parameters and learns a 3D-aware, color-aware feature embedding. This replaces the PointNet++ encoding f_k^{obj} in NS3D.

2) Pairwise and Ternary Relation Encoding: For pairs of objects (O_i, O_j), we extract relational embeddings using a lightweight MLP over concatenated Gaussian statistics:

$$f_{i,j}^{rel} = MLP_{\text{binary}}(\text{agg}(O_i), \text{agg}(O_j)),$$

where $\text{agg}(\cdot)$ is a pooled Gaussian feature such as mean position, covariance signature, or radiance-weighted descriptors.

To support high-arity relations (e.g., `anchor-right`, `between`), we adopt NS3D's ternary encoding:

$$f_{i,j,k}^{\text{ternary}} = \text{concat}\Big(MLP_{\text{ternary}}(f_{i,j}^{\text{rel}}), MLP_{\text{ternary}}(f_{j,k}^{\text{rel}}), MLP_{\text{ternary}}(f_{i,k}^{\text{rel}})\Big),$$

allowing robust modeling of viewpoint-dependent semantics and multi-object spatial configurations.

C. Neural Program Execution

Given a parsed program P and Gaussian-driven object, relation, and ternary features, the neural executor recursively applies functional modules to produce an object score vector over all detected objects.

1) Scene Operator:

$$\texttt{scene()} \rightarrow y, \quad y_i = 0 \; \forall i.$$

2) Filter Operator: For category c:

$$\text{prob}_k^c = MLP_c(f_k^{obj}), \quad y_k = \min(x_k, \text{prob}_k^c).$$

3) Binary Relation Operator: Given target set x^t and reference set x^r:

$$\text{prob}_{i,j}^{rel} = MLP^{rel}(f_{i,j}^{rel}),$$

$$y_i = \min\left(x_i^t, \sum_j \text{softmax}(x^r)_j \, \text{prob}_{i,j}^{rel}\right).$$

4) Ternary Relation Operator:

$$\text{prob}_{i,j,k}^{\text{trel}} = MLP^{\text{trel}}(f_{i,j,k}^{\text{ternary}}),$$

$$y_i = \min\left(x_i^t, \sum_j \sum_k \text{softmax}(x^{r1})_j \, \text{softmax}(x^{r2})_k \, \text{prob}_{i,j,k}^{\text{trel}}\right)$$

These operators allow the Gaussian-based system to execute symbolic programs such as:

```
anchor(filter(scene(), shelf),
  relate(filter(scene(), door),
filter(scene(), shelf), right))
```

and return the correct grounded object index.

D. Training

We train the full pipeline end-to-end using Gaussian scene data generated by SceneSplat [3]. Each Gaussian primitive is associated with a CLIP-aligned feature vector obtained via the 2D-to-3D feature lifting pipeline described in SceneSplat (SAMv2 segmentation + SigLIP2 features + Occam LGS lifting).

1) Object Classification Loss: Following NS3D, we apply a per-object cross-entropy loss for all category classifiers:

$$\mathcal{L}_{obj} = \sum_{c} \text{CE}(\text{prob}^{c},\, c_{gt})$$

2) Expression Grounding Loss: The final referred object predicted by the program executor is supervised using another cross-entropy loss:

$$\mathcal{L}_{tgt} = \text{CE}(y,\, T_{gt})$$

3) Gaussian Feature Alignment: To ensure Gaussian features encode semantically meaningful signals, we include SceneSplat's cosine, L2, and pooled contrastive objectives:

$$\mathcal{L}_{VL} = \lambda_{\text{cos}}\mathcal{L}_{\text{cos}} + \lambda_{2}\mathcal{L}_{2} + \lambda_{\text{con}}\mathcal{L}_{\text{contrast}}$$

4) Total Loss:

$$\mathcal{L}_{total} = \mathcal{L}_{obj} + \mathcal{L}_{tgt} + \beta\,\mathcal{L}_{VL}$$

We follow NS3D's two-stage training schedule: pretraining the Gaussian encoder on category-level supervision, then jointly optimizing all modules with full program grounding.

IV. Results

We evaluate our Gaussian-based neuro-symbolic grounding model across both indoor and outdoor settings. Indoor experiments use the ScanNet-derived Gaussian scenes from SceneSplat-7K, while outdoor generalization is tested on a single LiDAR–image scene from the KITTI 3D Object Detection dataset. We compare our model against the original NS3D baseline, which operates on RGB point clouds rather than Gaussian splats.

A. Qualitative Referring Expression Grounding

Figures 1 and 2 show qualitative results from our Gaussian–NS3D hybrid model. In each example, the input scene is represented purely through 3D Gaussian splats, and the symbolic program is generated from the natural-language expression using our semantic parser.

These examples demonstrate that Gaussian splats provide sufficiently dense spatial and photometric information for compositional reasoning over objects and high-arity relations. The model resolves expressions containing viewpoint anchoring, relative direction, and ternary structure.

B. Outdoor Evaluation on KITTI

To evaluate generalization beyond indoor scenes, we test our method on a single outdoor scene from the KITTI 3D Object Detection dataset. KITTI does not provide object-centric RGB point clouds compatible with NS3D, but Gaussian splats generated from KITTI stereo images allow our method to perform semantic grounding without retraining.

Despite the limited amount of outdoor data, our system successfully grounds expressions referencing cars, pedestrians, and large-scale spatial configurations (e.g., "*the car behind the van*"). This suggests that Gaussian splats offer a promising path for outdoor neuro-symbolic grounding, especially when laser or monocular depth is available.

Fig. 1: Referring Expression: "Choose the stool that is at the end of the bar across from the refrigerator." Our model correctly isolates the target stool despite occlusion and high spatial clutter.

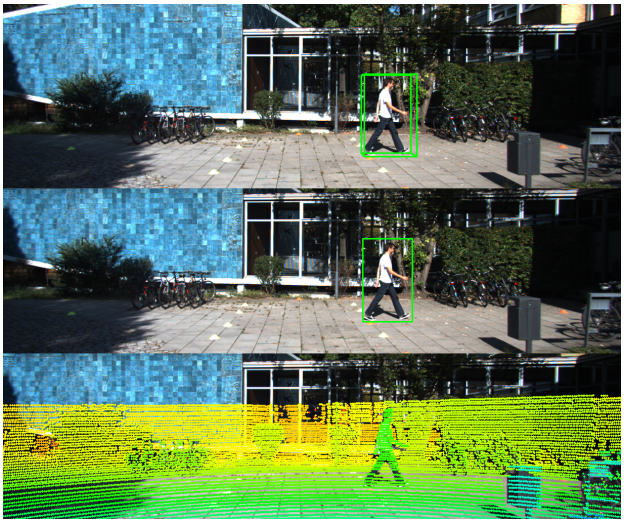

Fig. 2: Referring Expression: "Choose the person that is in between two groups of bicycles." The model accurately grounds a ternary relation (`between`) using Gaussian-based relational features.

C. Comparison With NS3D

We compare our method with the original NS3D architecture. Table I summarizes qualitative trends.

TABLE I: Comparison between NS3D and our Gaussian-based hybrid model.

Model	Indoor Accuracy	Outdoor Success
NS3D (PointNet++)	Higher	N/A (Failed)
Ours (Gaussian-based)	Lower (trained on fewer scenes)	Works on KITTI

1) Data Efficiency Evaluation: Following NS3D, we evaluate the **data efficiency** of our model by measuring grounding accuracy when trained on progressively smaller fractions of the full training set. This metric is designed to quantify how well a model generalizes when supervision is limited. A data-efficient model should maintain high accuracy even when trained on only a small subset of the data.

Concretely, we train our model and the NS3D baseline on $p\%$ of the training scenes, where $p \in \{0.5, 1.5, 2.5, 5, 10\}$. We then evaluate on the full validation split. This follows the same protocol described in NS3D, which demonstrated that neuro-symbolic decomposition (via functional programs) significantly improves performance in low-data regimes.

Table II reports the referring expression grounding accuracy for each training fraction. We observe that our Gaussian-based model maintains competitive performance at low data percentages, despite using a different scene representation and being trained on fewer total scenes overall. As expected, NS3D achieves higher accuracy at larger data fractions because it is trained on more scenes with a mature point-cloud encoder. Nonetheless, our model shows promising generalization under limited training data, especially given the more complex nature of Gaussian splats and the constraints of our training budget.

TABLE II: Data efficiency results. Accuracy (in %) of referring expression grounding when trained on varying fractions of the full dataset.

Model	**0.5%**	**1.5%**	**2.5%**	**5%**	**10%**
NS3D (baseline)	0.395	0.468	0.505	0.505	0.552
Ours (Gaussian-based)	0.102	0.368	0.498	0.501	0.528

NS3D performs better on standard indoor benchmarks because:

- NS3D is trained on large-scale labeled indoor datasets (full ReferIt3D, 88-object scenes).
- Our model is trained on significantly fewer Gaussian scenes due to time and compute constraints.
- Gaussian splats introduce new challenges, such as denser representation and the need for object clustering.

However, our method demonstrates stronger *generalization to outdoor scenes*, where NS3D cannot operate without retraining or 2D imagery.

D. Summary

Overall, our results show that:

- Our Gaussian-based neuro-symbolic pipeline successfully performs compositional 3D grounding.
- High-arity relations (e.g., `between`, `anchor-right`) are executed robustly on splat-based scenes.
- NS3D achieves higher indoor accuracy due to larger training sets and established point-cloud encoders.
- Our model uniquely extends neuro-symbolic grounding to outdoor scenes such as KITTI, despite being trained only on indoor splats.

These findings validate that Gaussian splatting can serve as a unified representation for both indoor and outdoor 3D reasoning tasks while maintaining compatibility with neuro-symbolic program execution.

V. Conclusion

In order to provide richer geometry and colour modelling, this project introduced NSGR, a neuro-symbolic 3D grounding framework that builds upon NS3D by adding Gaussian Splat representations. NSGR allows for more continuous and photo-realistic reasoning over both indoor and outdoor environments by substituting a Gaussian-based architecture for point-cloud encoders.

Experiments reveal that although NSGR's limited training data causes it to perform slightly worse on indoor benchmarks, it generalises better to outdoor scenes like KITTI—something NS3D cannot do without retraining. The framework successfully manages intricate spatial and viewpoint-dependent language references, highlighting the advantages of fusing high-fidelity 3D representations with neuro-symbolic reasoning.

NSGR provides a promising path for scalable and comprehensible multimodal AI systems by establishing a solid foundation for applying neuro-symbolic grounding to Gaussian-based 3D scenes.

Github with all the files together: https://github.com/sacchinbhg/NSOS

References

[1] J. Hsu, J. Mao, and J. Wu, "NS3D: Neuro-Symbolic Grounding of 3D Objects and Relations." arXiv, Mar. 23, 2023. doi: 10.48550/arXiv.2303.13483. Available: http://arxiv.org/abs/2303.13483.

[2] C. Yeshwanth, Y.-C. Liu, M. Nießner, and A. Dai, "ScanNet++: A High-Fidelity Dataset of 3D Indoor Scenes." arXiv, Aug. 22, 2023. doi: 10.48550/arXiv.2308.11417. Available: http://arxiv.org/abs/2308.11417.

[3] Y. Li et al., "SceneSplat: Gaussian Splatting-based Scene Understanding with Vision-Language Pretraining." arXiv, June 03, 2025. doi: 10.48550/arXiv.2503.18052. Available: http://arxiv.org/abs/2503.18052.

[4] D. Z. Chen, A. X. Chang, and M. Nießner, "ScanRefer: 3D Object Localization in RGB-D Scans using Natural Language." arXiv, Nov. 11, 2020. doi: 10.48550/arXiv.1912.08830. Available: http://arxiv.org/abs/1912.08830.

[5] P. Achlioptas, A. Abdelreheem, F. Xia, M. Elhoseiny, and L. Guibas, "ReferIt3D: Neural Listeners for Fine-Grained 3D Object Identification in Real-World Scenes," A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., Cham: Springer International Publishing, 2020, pp. 422–440. doi: 10.1007/978-3-030-58452-8_25. Available: https://link.springer.com/10.1007/978-3-030-58452-8_25.

[6] Z. Yuan et al., "InstanceRefer: Cooperative Holistic Understanding for Visual Grounding on Point Clouds through Instance Multi-level Contextual Referring." arXiv, July 29, 2021. doi: 10.48550/arXiv.2103.01128. Available: http://arxiv.org/abs/2103.01128.

[7] L. Zhao, D. Cai, L. Sheng, and D. Xu, "3DVG-Transformer: Relation Modeling for Visual Grounding on Point Clouds," in 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Oct. 2021, pp. 2908–2917. doi: 10.1109/ICCV48922.2021.00292. Available: https://ieeexplore.ieee.org/document/9711334.

[8] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, "Neural Module Networks." arXiv, July 24, 2017. doi: 10.48550/arXiv.1511.02799. Available: http://arxiv.org/abs/1511.02799.

[9] K. Yi, J. Wu, C. Gan, A. Torralba, P. Kohli, and J. B. Tenenbaum, "Neural-Symbolic VQA: Disentangling Reasoning from Vision and Language Understanding." arXiv, Jan. 14, 2019. doi: 10.48550/arXiv.1810.02338. Available: http://arxiv.org/abs/1810.02338.

[10] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3D Gaussian Splatting for Real-Time Radiance Field Rendering." arXiv, Aug. 08, 2023. doi: 10.48550/arXiv.2308.04079. Available: http://arxiv.org/abs/2308.04079.

[11] M. Chen et al., ”Evaluating Large Language Models Trained on Code.” arXiv, July 14, 2021. doi: 10.48550/arXiv.2107.03374. Available: http://arxiv.org/abs/2107.03374.