Exploring Zero-Shot Object Recognition in Language-Grounded 3D Occupancy Prediction Models

Eunjae Hong
Seoul National University
eunjae.hong@snu.ac.kr

Eugene Sohn
Seoul National University
qeugene@snu.ac.kr

Jonghem Youn
Seoul National University
jonghm@snu.ac.kr

Jungin Hong
Seoul National University
junginh@snu.ac.kr

Abstract

Vision-centric 3D occupancy prediction provides a detailed representation of driving scenes, but standard models are limited to a fixed closed vocabulary. We study how CLIP-aligned visual semantics can be incorporated into a TPVFormer-based occupancy pipeline for zero-shot object recognition. Specifically, we compare three alignment strategies that operate at different stages of the 2D-to-3D lifting pipeline: projection-based alignment of FPN features, FPN fine-tuning with patch-level CLIP feature distillation, and replacement of the ResNet image encoder with a CLIP-pretrained ResNet backbone. To train the explicit alignment strategies, we construct patch-level CLIP visual supervision from nuScenes images using the FPN grid and effective receptive field sizes, avoiding external open-vocabulary segmenters or rendering-based supervision. The aligned image-side modules are then used to retrain TPVFormer with the original occupancy objective, allowing the downstream 3D representation to adapt to the modified visual features. Experiments under both the original closed-set occupancy setting and an extended CLIP text-querying protocol provide an initial comparison of these strategies, suggesting that the choice of alignment stage influences the behavior of the resulting occupancy model.

1. Introduction

Modern autonomous driving systems are shifting toward dense, vision-centric 3D semantic occupancy prediction moving beyond sparse object-level bounding boxes [3, 15]. By partitioning the 3D space into a volumetric representation, this formulation captures both structural geometry and semantics of complex environments. Although modern frameworks efficiently lift multi-view 2D features into 3D spaces via spatial cross-attention [5, 10], they operate under a strict closed-vocabulary assumption. Relying on fixed, predefined taxonomies, these models fail to recognize out-of-vocabulary (OOV) objects—such as rare construction vehicles or road debris—posing critical safety risks.

To bypass fixed-label limitations, recent methods integrate 2D Vision-Language Models (VLMs) [7, 13] into open-vocabulary 3D perception [6, 12]. These methods transfer pre-trained multi-modal knowledge into 3D space to achieve open-world zero-shot understanding [14]. Existing strategies conventionally distill dense pixel-level features from off-the-shelf 2D segmenters [4] or employ self-supervised view synthesis to mitigate depth ambiguity [1]. However, pixel-level distillation often suffers from inaccurate 2D segmentation boundaries, whereas view synthesis incurs high computational costs [8, 9]. More importantly, existing approaches typically apply cross-modal distillation without fully accounting for the structural nuances of the 3D representation pipeline. This leaves a fundamental question under-explored: how and at which stage of the 2D-to-3D lifting process should language alignment be established to maximize open-vocabulary capability?

We therefore investigate language alignment at different stages of the 2D-to-3D lifting pipeline for zero-shot voxel querying and open-vocabulary 3D occupancy prediction. Rather than focusing only on whether CLIP features can be injected into the model, we examine where such alignment should be introduced. In particular, we compare three image-side strategies: projecting FPN features into the CLIP space, distilling CLIP visual features into the FPN neck, and replacing the image encoder with a CLIP-pretrained backbone.

To support the explicit alignment strategies, we construct patch-level CLIP visual supervision from nuScenes camera images. The supervision is defined on the FPN grid using ERF-calibrated image crops, providing a lightweight alter-

native to external open-vocabulary segmenters or rendering-based supervision. Using these aligned image-side modules, we retrain TPVFormer with the original occupancy objective and evaluate both closed-set occupancy performance and CLIP text-based voxel querying behavior.

Our main contributions are summarized as follows:

- We investigate CLIP-based language alignment for zero-shot object recognition in TPVFormer-based 3D occupancy prediction, focusing on where language-compatible visual features should be introduced in the 2D-to-3D lifting pipeline.
- We compare three image-side alignment strategies—FPN feature projection, FPN-level CLIP distillation, and CLIP-pretrained encoder replacement—and construct an ERF-calibrated patch-level CLIP supervision scheme on nuScenes images without relying on external open-vocabulary segmenters or rendering-based losses.
- We evaluate the resulting models under both closed-set occupancy prediction and an extended CLIP text-querying protocol, showing that the alignment position affects the trade-off between in-vocabulary stability and openness to OOV categories, while highlighting the limitations of direct CLIP-based voxel querying for reliable open-vocabulary 3D occupancy.

2. Related Work

Vision-centric 3D Occupancy.

Vision-centric 3D semantic occupancy prediction aims to reconstruct the continuous volumetric state of the environment using camera-only inputs. Early frameworks established dense 2D-to-3D projection pipelines using 3D CNNs [3]. To capture spatial and temporal context, BEVFormer [10] introduced a transformer-based architecture to construct bird's-eye-view (BEV) representations. Subsequent studies extended features into multi-scale voxel queries [15] or efficient alternative representations, such as Tri-Perspective View (TPV) [5] and implicit self-supervised rendering [11], to mitigate the computational overhead of dense grids. Despite their structural efficiency, these networks strictly operate under a closed-vocabulary assumption. Their reliance on heavy, predefined 3D semantic annotations restricts deployment in open-world scenarios where out-of-vocabulary (OOV) objects are inevitable.

Open-Vocabulary 3D Perception.

Open-vocabulary 3D perception has emerged to overcome the constraints of closed-set assumption, primarily by establishing cross-modal alignment between 3D representations and pre-trained VLMs [13]. Early methods leveraged 2D open-vocabulary segmenters to distill pixel-level features into point clouds or voxels [14], whereas recent frameworks utilize self-supervised view synthesis to resolve depth ambiguity during cross-modal distillation [1]. However, these methods focus on *whether* to distill VLM features, leaving a fundamental structural question unanswered: *at which stage of the 2D-to-3D lifting pipeline should the multi-modal alignment occur?* Unlike existing works that apply distillation in an ad-hoc or purely end-to-end manner, we conduct a systematic comparison of three architectural alignment strategies: post-hoc feature alignment, FPN-level feature distillation, and CLIP-pretrained visual encoder replacement, to identify where language-aware representations are most effectively introduced into 3D occupancy prediction.

Dense Image-Text Supervision in 3D.

To effectively ground language embeddings into dense 3D spaces, obtaining high-quality multi-modal supervision is crucial. Current mainstream approaches primarily fall into two categories: pixel-level feature distillation from off-the-shelf 2D open-vocabulary segmenters [4] or volume rendering-based cross-modal alignment via differentiable ray-marching [8, 9]. However, pixel-level distillation often introduces severe boundary noise from 2D predictors, while volume rendering suffers from excessive computational and optimization overhead. On the other hand, although patch-level or region-level supervision has been explored in 2D vision-language learning to capture localized semantics [16], its adaptation to 3D driving scenarios remains under-explored due to depth ambiguity and perspective distortion. Distinct from these methods, our framework introduces an efficient patch-level CLIP visual supervision scheme on the image side. Specifically, we define patch locations on the FPN grid, crop the corresponding nuScenes camera image regions using ERF-calibrated crop sizes, and use a frozen CLIP image encoder to obtain visual feature targets. This design bypasses external 2D open-vocabulary segmenters and rendering-based supervision, while providing a lightweight way to inject CLIP-compatible visual semantics before TPV lifting.

3. Method

3.1. Unified Framework and Design Principle

We investigate how language-aware visual semantics can be introduced into camera-based 3D occupancy prediction for zero-shot object recognition. Our study is based on a common design principle: image features should be aligned with the CLIP feature space before or during their transition into the voxel representation. Since TPVFormer constructs 3D scene representations by aggregating multi-view 2D image features through cross-attention, the semantic quality of the image features directly affects the resulting voxel features.

Under this principle, we study three independent alignment strategies. Figure 2 illustrates the overall TPVFormer-based framework and the insertion points of the three alignment strategies within the 2D-to-3D lifting pipeline. The

first strategy adds a lightweight projection head between the FPN and TPVFormer cross-attention. The second strategy fine-tunes the FPN neck using an auxiliary patch-level CLIP distillation loss while keeping the ResNet image backbone fixed. The third strategy replaces the ResNet image backbone with a CLIP-pretrained encoder.

These strategies modify different parts of the image feature extraction pipeline to induce language-compatible 3D features. They are not different querying mechanisms; instead, they provide alternative ways of preparing TPV-derived features for a shared zero-shot voxel querying protocol based on CLIP text embeddings. This formulation allows us to analyze where language alignment is most effective: as a post-hoc projection on top of conventional image features, as direct supervision for the FPN feature maps, or as a vision-language pretrained image encoder.

3.2. Patch-Level CLIP Feature Extraction from nuScenes

To train the explicit image feature alignment strategies, we extract patch-level CLIP visual features from nuScenes camera images using the FPN feature grid and ERF-calibrated crop sizes. nuScenes provides synchronized multi-view camera images, LiDAR point clouds, 3D bounding box annotations, and accurate camera–LiDAR calibration parameters.

These patches are defined on the FPN feature map grid used by TPVFormer. The FPN backbone produces feature maps at four levels: P3 (116×200), P4 (58×100), P5 (29×50), and P6 (15×25), yielding 23,200, 5,800, 1,450, and 375 patch positions per image per level, respectively. For each grid cell, the patch center is obtained by uniformly interpolating positions within the valid region, inset by half the crop size on each side to prevent out-of-bounds sampling. The crop size at each level is set to the two-sigma effective receptive field (ERF): 59.3 px for P3, 257.5 px for P4, 314.8 px for P5, and 331.7 px for P6, rounded to the nearest even integer. Each cropped patch is resized to 224×224 and passed through the CLIP image encoder (ViT-B/32), producing a 512-dimensional ℓ_2-normalized feature vector stored in float16.

Across all four levels, a single camera image yields 30,825 CLIP feature vectors; with six cameras per keyframe, this amounts to 184,950 vectors per sample. Due to practical storage constraints, we do not process the full 700-scene nuScenes training split. Instead, we select the top 100 scenes from the 700-scene nuScenes training split, ranked by a composite score over three criteria: category diversity (number of unique object categories per scene), small-object ratio (fraction of instances with 3D bounding box volume below $2.0\,\mathrm{m}^3$), and semantic ambiguity (number of confusable category pairs co-occurring in the scene), weighted 0.4, 0.4, and 0.2 respectively. Figure 1 visualizes

highest score (a) scene-0980 lowest score (b) scene-1079

Figure 1. Representative frames from the highest- and lowest-scoring scenes under our composite selection criterion. (a) The highest-scoring scene depicts a construction site with small objects including pedestrians, barriers, traffic cones, and construction vehicles, reflecting high category diversity and small-object ratio. (b) The lowest-scoring scene consists predominantly of open road with only a few large vehicles, yielding low diversity and minimal semantic ambiguity.

representative frames from the highest- and lowest-scoring scenes under this selection criterion, illustrating that the selected scenes contain richer object diversity and more ambiguous small-scale objects.

3.3. Strategy I: Projection-Based Feature Alignment

This strategy aligns FPN image features with the CLIP visual embedding space by training a lightweight projection head on patch-level visual supervision. Rather than modifying the 3D occupancy objective, the projection head is trained independently using pre-extracted CLIP visual features as targets, and its weights are subsequently transferred into the full TPVFormer pipeline.

Projection head architecture. Let $F_i^{(l)} \in \mathbb{R}^{H_l \times W_l \times C}$ denote the FPN feature map for camera i at level $l \in \{P3, P4, P5, P6\}$, where $C = 256$. A projection head h_θ maps each spatial feature vector to the CLIP embedding dimension $D = 512$:

$$\tilde{f} = h_\theta(f), \quad \tilde{f} \in \mathbb{R}^D, \tag{1}$$

where h_θ is an MLP with hidden dimension $H = 512$ and depth $K = 2$ (658,944 parameters).

Training supervision. For each training sample, CLIP visual features are pre-extracted from the image patches using a frozen CLIP image encoder (ViT-B/32) and stored per FPN spatial position. At each FPN level l, $M = 32{,}768$ positions are randomly subsampled from all spatial locations across the batch and cameras. For each selected position m, let $v_m \in \mathbb{R}^D$ denote the corresponding pre-extracted CLIP visual feature. The training objective minimizes the cosine

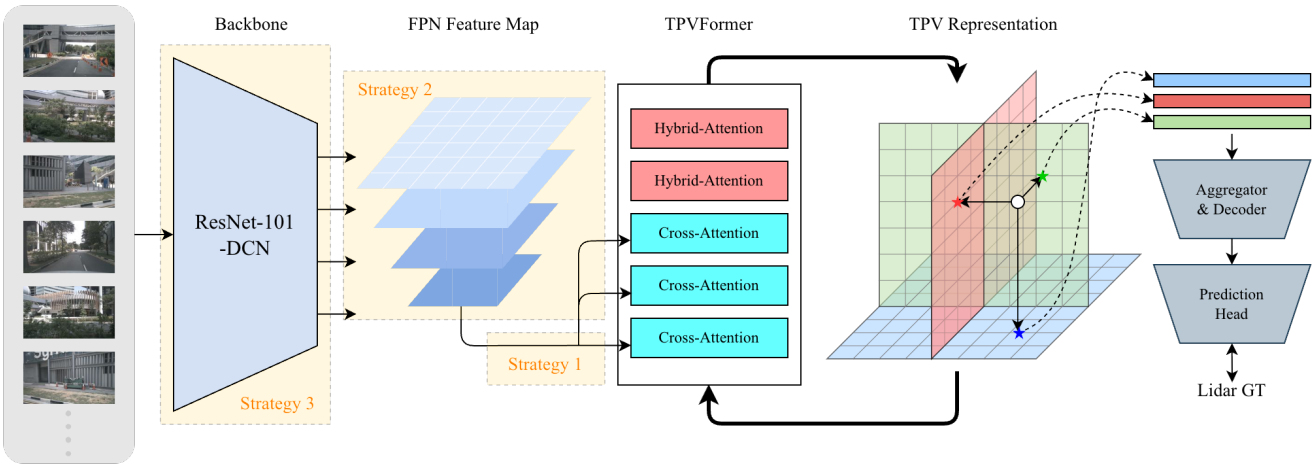

Figure 2. Overview of the proposed CLIP-aligned TPV framework. Three image-side alignment strategies are introduced to study how CLIP-compatible visual features affect TPV-derived occupancy representations under text-based voxel querying.

distance between the L2-normalized projection and the L2-normalized CLIP visual target:

$$z_m = \frac{h_\theta(f_m)}{\|h_\theta(f_m)\|_2}, \qquad \hat{v}_m = \frac{v_m}{\|v_m\|_2}, \tag{2}$$

$$\mathcal{L}_{\text{align}} = 1 - \frac{1}{M}\sum_{m=1}^{M} z_m^\top \hat{v}_m. \tag{3}$$

The loss is averaged over all subsampled positions and FPN levels, and all spatial positions are treated equally regardless of their semantic class.

Integration into TPVFormer. After training h_θ, we introduce an FPNBridge module that combines the pre-trained projection h_θ with a learned linear reduction layer $r_\phi : \mathbb{R}^D \to \mathbb{R}^C$. The full pipeline becomes

$$\hat{F}_i^{(l)} = r_\phi(h_\theta(F_i^{(l)})), \quad \hat{F}_i^{(l)} \in \mathbb{R}^{H_l \times W_l \times C}. \tag{4}$$

The reduced features $\{\hat{F}_i^{(l)}\}$ replace the original FPN outputs as keys and values in TPVFormer cross-attention. Rather than training h_θ jointly with the occupancy objective, we pre-train it independently so that the CLIP alignment signal is not diluted by the long gradient path through TPV lifting and voxel decoding. The reduction layer r_ϕ is randomly initialized and trained jointly with the occupancy objective, while h_θ is initialized from the pre-trained projection weights. This strategy tests whether CLIP-aligned image features, when transferred to the 3D representation through TPV lifting, improve open-vocabulary recognition. The second stage follows the original TPVFormer training protocol (AdamW, lr=2×10^{-4}, weight decay 0.01, 24 epochs), with the image backbone, FPN neck, and FPN-Bridge using a reduced learning rate ($\times0.1$) to preserve pre-trained weights.

3.4. Strategy II: FPN Fine-tuning with CLIP Feature Distillation

The second strategy aligns the image features consumed by TPVFormer by fine-tuning the FPN neck with patch-level CLIP visual supervision. Unlike Strategy I, which learns an additional projection head on top of fixed ResNet-FPN features, Strategy II directly updates the FPN feature extractor so that the multi-scale feature maps passed to TPVFormer become closer to the CLIP visual feature space. The ResNet backbone is kept frozen to preserve its pretrained low- and mid-level visual representations, while the FPN neck is updated because it directly produces the keys and values used in TPVFormer cross-attention.

We use the same patch-level CLIP visual supervision described in Sec. 3.2. For each multi-view training sample, the frozen ResNet backbone and trainable FPN neck produce multi-scale feature maps at levels $l \in \{P_3, P_4, P_5, P_6\}$:

$$F^l \in \mathbb{R}^{B\times N\times C\times H_l\times W_l}, \tag{5}$$

where B is the batch size, N is the number of camera views, and $C = 256$ is the FPN channel dimension. For each spatial location u at level l, we use the pre-extracted CLIP visual feature $q_u^l \in \mathbb{R}^{512}$ from the corresponding ERF-calibrated image patch as the target.

Since the FPN and CLIP feature dimensions differ, we attach a lightweight auxiliary adapter $g_\phi : \mathbb{R}^{256} \to \mathbb{R}^{512}$ only for the distillation loss. For a selected location u, the normalized prediction and target are

$$\hat{z}_u^l = \frac{g_\phi(F_u^l)}{\|g_\phi(F_u^l)\|_2}, \qquad \hat{q}_u^l = \frac{q_u^l}{\|q_u^l\|_2}. \tag{6}$$

The FPN distillation objective is the average cosine distance

over the selected levels and patch locations:

$$\mathcal{L}_{clip} = \frac{1}{|S|} \sum_{l \in S} \frac{1}{|\Omega_l|} \sum_{u \in \Omega_l} \left(1 - \hat{z}_u^l \cdot \hat{q}_u^l\right). \tag{7}$$

During this stage, gradients from $\mathcal{L}_{clip}$ update only the FPN neck and the auxiliary adapter. The ResNet backbone, TPV lifting module, TPV aggregation module, and occupancy decoder are kept frozen. Thus, this stage performs image-side feature alignment rather than occupancy training.

After distillation, the auxiliary adapter is discarded. Only the updated FPN weights are transferred to the full TPVFormer pipeline, which is then trained using the original voxel-level occupancy objective:

$$\mathcal{L}_{TPV} = \mathcal{L}_{occ} = \mathcal{L}_{CE}^{voxel} + \mathcal{L}_{Lovasz}^{voxel}. \tag{8}$$

No CLIP feature files are loaded during this downstream occupancy training stage, and no additional CLIP loss is added. At inference time, the pipeline remains identical to the original TPVFormer pipeline except that it uses the CLIP-distilled FPN neck.

3.5. Strategy III: CLIP-RN101 Image Encoder

The third strategy replaces the conventional ImageNet-pretrained ResNet backbone with CLIP-RN101. Unlike Strategies I and II, which introduce CLIP alignment through an additional projection head or auxiliary loss, this strategy changes only the image encoder and relies on the vision-language prior already learned during CLIP pretraining. The goal is to test whether a CLIP-pretrained visual backbone can provide more language-compatible image features for downstream 3D occupancy prediction.

Since CLIP-RN101 is based on the ResNet architecture, we can extract intermediate feature maps from multiple stages of the encoder and construct an FPN-style multi-scale representation:

$$\{C_3^{\text{clip}}, C_4^{\text{clip}}, C_5^{\text{clip}}\} = \text{CLIP-RN101}(I_i), \tag{9}$$

$$\{P_3, P_4, P_5, P_6\} = \text{FPN}(\{C_3^{\text{clip}}, C_4^{\text{clip}}, C_5^{\text{clip}}\}). \tag{10}$$

The resulting multi-scale image features are then fed into TPVFormer cross-attention in the same manner as the original ResNet-FPN features. In this strategy, no additional patch-level CLIP visual alignment loss is introduced. Instead, language alignment is incorporated through the CLIP-pretrained image encoder itself. By keeping the alignment mechanism implicit in the pretrained encoder, this strategy provides a complementary comparison to the explicit alignment objectives used in Strategies I and II.

4. Experiments

4.1. Experimental Setup

Dataset. We use the nuScenes v1.0 trainval dataset [2] in the camera-only setting, where each keyframe contains six synchronized camera images and the corresponding camera–LiDAR calibration parameters. We follow the official nuScenes split, using 700 scenes for training and 150 for evaluation. The preprocessing pipeline follows TPVFormer, sharing the same camera views, voxel range, and semantic labels.

Label spaces. The supervised occupancy task follows the standard nuScenes learning map used by TPVFormer. This map converts the original nuScenes raw labels into 16 valid semantic classes: barrier, bicycle, bus, car, construction vehicle, motorcycle, pedestrian, traffic cone, trailer, truck, driveable surface, other flat, sidewalk, terrain, manmade, and vegetation.

In addition to the 16-class closed-set label space, we define a 25-class vocabulary by appending nine raw nuScenes categories that the standard learning map collapses into the ignore label: animal, personal mobility rider, stroller, wheelchair, debris, pushable/pullable object, bicycle rack, ambulance, and police car. This 25-class vocabulary is not arbitrary; it is derived from nuScenes categories that exist in the original annotation space but are excluded from closed-set training.

CLIP supervision. For methods requiring image-level CLIP targets, we precompute CLIP (ViT-B/32) features on 100 selected training scenes (4,002 keyframes), chosen to maximize category diversity, small-object ratio, and semantic ambiguity. This fixed subset is used consistently across all alignment strategies.

Base occupancy model. All experiments build on TPVFormer [5] as the shared backbone. Unless otherwise stated, the base model follows the TPVFormer occupancy configuration with a ResNet-101-DCN image backbone, an FPN image neck, four 256-dimensional feature levels, and six camera views. The TPV representation uses a $100 \times 100 \times 8$ grid over the spatial range $[-51.2, 51.2]$ m in the horizontal plane and $[-5.0, 3.0]$ m in height.

4.2. Evaluation Metrics

For each validation sample, we run the occupancy model in evaluation mode and extract the pre-classifier feature from the TPV aggregator output. Since the decoder output is defined on the TPV voxel grid, we sample the corresponding voxel feature at each LiDAR point location using the point-to-voxel grid indices provided by the dataset. The metrics are therefore accumulated at point locations, while the queried representation is still the TPV-derived voxel feature.

For CLIP text querying, we map the TPV-derived voxel feature to the 512-dimensional CLIP text embedding dimension using a fixed no-training random Gaussian projection. Specifically, the decoder voxel feature $f_i \in \mathbb{R}^{256}$ is linearly projected using a random matrix $W \in \mathbb{R}^{256 \times 512}$

generated with seed 42:

$$\tilde{f}_i = f_i W. \tag{11}$$

The columns of W are normalized before projection, and the projected feature is then ℓ_2-normalized:

$$\hat{f}_i = \frac{\tilde{f}_i}{\|\tilde{f}_i\|_2}. \tag{12}$$

This projection is shared by all methods and is not learned. For a class name c, the frozen CLIP text encoder produces a normalized text embedding t_c using the prompt template *"a photo of a {class}"*. The class score is computed by cosine similarity:

$$s_{i,c} = \frac{\hat{f}_i^\top t_c}{\tau}, \tag{13}$$

where $\hat{f}_i$ is the normalized point feature after projection and $\tau = 1.0$ is the evaluation temperature. We compute predictions under two vocabularies:

$$\hat{y}_i^{(K)} = \arg\max_{c \in \mathcal{C}_K} s_{i,c}, \quad K \in \{16, 25\}. \tag{14}$$

Here, $\mathcal{C}_{16}$ is the standard 16-class nuScenes occupancy vocabulary, and $\mathcal{C}_{25}$ is the extended vocabulary that appends the nine raw nuScenes categories described in Sec. 4.1.

For a vocabulary $\mathcal{C}_K$, we compute per-class intersection-over-union by excluding the ignore label:

$$\mathrm{IoU}_c = \frac{TP_c}{TP_c + FP_c + FN_c}, \quad c \in \mathcal{V}_K, \tag{15}$$

where $\mathcal{V}_K$ is the set of evaluated non-ignore classes under vocabulary $\mathcal{C}_K$. Classes that are absent from both predictions and ground truth are not included in the average. The mean IoU is

$$\mathrm{mIoU}_K = \frac{1}{|\mathcal{V}_K|} \sum_{c \in \mathcal{V}_K} \mathrm{IoU}_c, \quad K \in \{16, 25\}. \tag{16}$$

Here, mIoU_{16} measures closed-set occupancy quality with the standard nuScenes classes, while mIoU_{25} applies the same formula over the extended 25-class vocabulary. Since the standard occupancy ground truth remains mapped to the 16-class label space, predicting appended OOV names for ordinary in-vocabulary points lowers the 25-class score. We report this degradation as

$$\Delta\mathrm{mIoU} = \mathrm{mIoU}_{16} - \mathrm{mIoU}_{25}. \tag{17}$$

To directly quantify this failure mode, we also measure in-vocabulary to OOV leakage. Let $\mathcal{C}_{in}$ be the 16 valid nuScenes classes and $\mathcal{C}_{out}$ be the nine appended classes in the 25-class vocabulary. The leakage rate is

$$\mathrm{Leak}_{in \to out} = \frac{|\{i \mid y_i \in \mathcal{C}_{in},\ \hat{y}_i^{(25)} \in \mathcal{C}_{out}\}|}{|\{i \mid y_i \in \mathcal{C}_{in}\}|}. \tag{18}$$

This metric is important because an open-vocabulary model should not assign ordinary in-vocabulary objects to newly added class names too often. For points whose raw ground-truth labels belong to the appended OOV categories, we further evaluate OOV recognition using two complementary metrics. Let C_{out} denote the nine appended OOV categories, and let $N_{\mathrm{OOV}} = |\{i \mid y_i \in C_{\mathrm{out}}\}|$ be the number of points whose raw labels belong to these categories.

OOV-Any measures whether such a point is assigned to any appended OOV category:

$$\text{OOV-Any} = \frac{1}{N_{\mathrm{OOV}}} \sum_{i: y_i \in C_{\mathrm{out}}} \mathbf{1}\left[\hat{y}_i^{(25)} \in C_{\mathrm{out}}\right]. \tag{19}$$

This metric captures whether the model can separate excluded raw categories from the original in-vocabulary classes.

OOV-Exact measures whether the predicted OOV category exactly matches the raw ground-truth category:

$$\text{OOV-Exact} = \frac{1}{N_{\mathrm{OOV}}} \sum_{i: y_i \in C_{\mathrm{out}}} \mathbf{1}\left[\hat{y}_i^{(25)} = y_i\right]. \tag{20}$$

Unlike OOV-Any, this metric evaluates class-level zero-shot recognition among the appended OOV categories. If raw OOV labels are unavailable after applying the standard nuScenes learning map, both metrics are reported as not applicable.

4.3. Effect of Alignment Position on Occupancy Prediction

Table 2 reports the closed-set validation results. Introducing CLIP supervision does not harm—and for the explicit alignment strategies even slightly improves—standard occupancy prediction: Strategy I raises the point-level mIoU from 26.839 to 27.983 and the voxel-level mIoU from 52.059 to 53.477, while Strategy II attains the best voxel-level mIoU of 53.876. This result suggests that pre-aligning the image features toward the CLIP space does not conflict with the standard occupancy objective and may even provide a slightly more favorable starting point for it. In contrast, Strategy III is the only variant that degrades occupancy quality (voxel-level mIoU 44.778, loss 0.756). One possible explanation is that replacing the entire image encoder with CLIP-RN101 alters the feature distribution fed into TPVFormer, and the resulting features may be less compatible with the multi-scale FPN representation and TPV-based 2D-to-3D lifting than the original ImageNet-pretrained backbone.

We next turn to open-vocabulary behavior under the CLIP text-querying protocol, where TPV-derived voxel features are matched directly against CLIP text embeddings of class names. Table 1 reports the results under the 16-class and the extended 25-class vocabularies. Two trends stand

Table 1. Quantitative comparison of the baseline TPVFormer and three language-alignment strategies under the CLIP text-querying protocol. All mIoU values in this table are CLIP-query mIoU. $mIoU_{16}$ evaluates the original 16-class closed-set vocabulary, while $mIoU_{25}$ evaluates the extended vocabulary with nine appended OOV categories. $\Delta mIoU$ denotes the drop from $mIoU_{16}$ to $mIoU_{25}$, and In→OOV measures the fraction of in-vocabulary ground-truth points predicted as appended OOV categories. OOV-Any measures whether an OOV point is assigned to any appended OOV category, while OOV-Exact measures exact class-level OOV recognition among the appended categories.

Method	$mIoU_{16}\uparrow$	$mIoU_{25}\uparrow$	$\Delta mIoU\downarrow$	In→OOV↓	OOV Any↑	OOV Exact↑
TPVFormer baseline	2.49	1.60	0.89	13.38	13.61	0.74
Strategy I	2.34	0.59	1.74	40.96	**61.39**	0.89
Strategy II	1.38	0.91	**0.47**	**2.74**	10.69	0.10
Strategy III	1.12	0.64	0.49	21.92	16.97	**1.43**

Table 2. Validation performance comparison across TPVFormer and the three alignment strategies.

Method	$\text{mIoU}_{\text{pts}}\uparrow$	$\text{mIoU}_{\text{vox}}\uparrow$	Loss ↓
TPVFormer baseline	26.839	52.059	0.708
Strategy I	**27.983**	53.477	**0.700**
Strategy II	27.293	**53.876**	0.702
Strategy III	25.634	44.778	0.756

out. First, the absolute open-vocabulary mIoU remains low for all methods ($mIoU_{16} \leq 2.49$), indicating that voxel features optimized for occupancy are still only weakly comparable with CLIP text embeddings. Second, and more importantly, the alignment strategies do not so much raise this absolute score as reshape the predicted label distribution, trading in-vocabulary stability against openness to the appended OOV names: Strategy I is the most open, reaching the highest OOV recognition (OOV-Any $= 61.39$) but also the largest in-vocabulary-to-OOV leakage (40.96%); Strategy II is the most conservative, with the lowest leakage (2.74%) but weak OOV recognition; and Strategy III stays close to the baseline in mIoU while roughly doubling the leakage.

We attribute this persistently low absolute mIoU to several main factors. First, the CLIP alignment is established in the 2D feature space at the front of the pipeline, but the aligned signal must then pass through the heavy TPVFormer cross-attention, the TPV aggregator, and the occupancy decoder before the resulting voxel feature is mapped to the CLIP text dimension by the fixed random projection used for evaluation; this long, non-linear path likely dilutes much of the CLIP-compatible semantics injected upstream. Second, the patch-level CLIP supervision itself is imperfect: the cropped patches are small relative to the full image and may not carry sufficient object-level semantics, an issue that is most severe at the finest level P3 and is compounded by the dimensional gap between the 256-channel FPN features and the 512-dimensional CLIP space. Third, owing to storage constraints we extract supervision from only 100 selected scenes, which introduces a scene-selection bias that limits the diversity of the learned alignment. Together, these factors offer a plausible explanation for why structural alignment reshapes the predicted distribution well before it can yield strong absolute zero-shot accuracy.

In summary, our experiments show that the alignment strategies preserve occupancy prediction performance while measurably changing the distribution of the predicted features. Compared with the baseline, the explicit strategies leave the closed-set mIoU essentially intact yet clearly redistribute how points are assigned under the CLIP text-querying protocol, as reflected in the shifts in OOV recognition and in-vocabulary-to-OOV leakage in Table 1. However, it remains uncertain whether this distributional change actually corresponds to alignment in the intended CLIP direction. The low absolute open-vocabulary mIoU, together with the imbalance between OOV recognition and leakage, suggests that the observed shift is not necessarily a faithful movement toward the CLIP semantic space and may instead reflect an uncalibrated or partially spurious change. Establishing whether the feature distribution genuinely moves toward CLIP-aligned semantics, and identifying the conditions under which it does, therefore requires further investigation—for instance through stronger patch-level supervision, broader scene coverage, and more direct measures of feature–text alignment.

Figure 3 further illustrates this behavior qualitatively. When visible points are projected onto the front camera image and colored by their predicted 25-class labels, the prediction map shows that the model roughly preserves the spatial structure of the scene and captures where objects or occupied regions are located. However, the assigned labels are often inconsistent, indicating that the model retains coarse objectness or occupancy cues but fails to reliably discriminate the corresponding semantic categories under CLIP text querying. This supports our observation that CLIP supervision affects the semantic behavior of the representation while largely preserving the original occupancy prediction performance shown in Table 2.

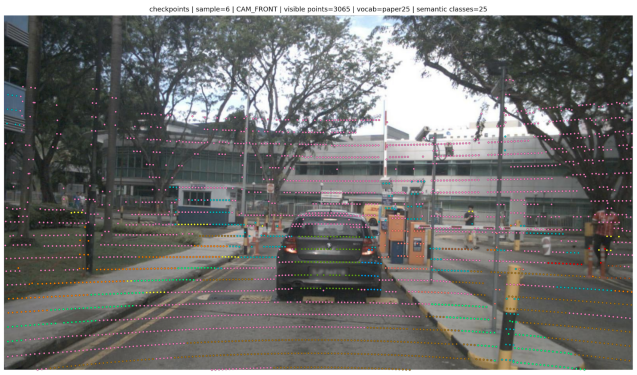

Figure 3. Qualitative visualization of 25-class CLIP text querying. Visible points projected onto the front camera image are colored according to the semantic class predicted by querying TPV-derived occupancy features with CLIP text embeddings. The result preserves the coarse spatial layout of occupied regions, but semantic labels are often noisy and locally inconsistent, suggesting that direct CLIP text querying captures occupancy structure better than fine-grained class discrimination.

5. Conclusion

In this work, we explored CLIP-based language alignment for zero-shot object recognition in TPVFormer-based 3D occupancy prediction. We compared three image-side alignment strategies—feature projection, FPN-level CLIP distillation, and CLIP-pretrained encoder replacement—to analyze how language-compatible visual semantics can be introduced into the 2D-to-3D lifting pipeline.

Our experiments show that CLIP-oriented image-side alignment can be incorporated into TPVFormer without substantially degrading closed-set occupancy prediction. In particular, the explicit alignment strategies preserve, and in some cases slightly improve, standard occupancy performance, suggesting that language-aware visual supervision can coexist with the original voxel-level objective. Under the extended CLIP text-querying protocol, the alignment strategies further change the prediction behavior of TPV-derived voxel features, indicating that the injected CLIP semantics are at least partially reflected in the final 3D representation.

At the same time, the resulting open-vocabulary recognition remains limited. The observed changes mainly appear as shifts in the prediction distribution, with different strategies showing different trade-offs between in-vocabulary stability and openness to appended OOV categories. These results suggest that image-side alignment is a useful starting point, but reliable open-vocabulary 3D occupancy prediction likely requires additional mechanisms to preserve and calibrate language-compatible features through TPV lifting, aggregation, and decoding.

Overall, our study provides an initial empirical basis for understanding how image-side CLIP alignment affects TPV-based 3D occupancy representations, and points to the need for stronger feature preservation and calibration mechanisms for future open-vocabulary 3D scene understanding.

References

[1] Simon Boeder, Fabian Gigengack, and Benjamin Risse. Langocc: Open vocabulary occupancy estimation via volume rendering. In *2025 International Conference on 3D Vision (3DV)*, pages 200–210. IEEE Computer Society, 2025. 1, 2

[2] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 5

[3] Anh-Quan Cao and Raoul de Charette. Monoscene: Monocular 3d semantic scene completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3991–4001, 2022. 1, 2

[4] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling open-vocabulary image segmentation with image-level labels. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 540–557, 2022. 1, 2

[5] Yuanhui Huang, Wenzhao Zheng, Yunpeng Zhang, Jie Zhou, and Jiwen Lu. Tri-perspective view for vision-based 3d semantic occupancy prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9223–9232, 2023. 1, 2, 5

[6] Krishna Murthy Jatavallabhula, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Alaa Maalouf, Shuang Li, Ganesh Subramanian Iyer, Nikhil Varma Keetha, Ayush Tewari, Joshua B. Tenenbaum, Celso M de Melo, Madhava Krishna, Liam Paull, Florian Shkurti, and Antonio Torralba. Conceptfusion: Open-set multimodal 3d mapping. In *ICRA2023 Workshop on Pretraining for Robotics (PT4R)*, 2023. 1

[7] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021. 1

[8] Haochen Jiang, Yueming Xu, Yihan Zeng, Hang Xu, Wei Zhang, Jianfeng Feng, and Li Zhang. Openocc: Open vocabulary 3d scene reconstruction via occupancy representation. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1145–1152, 2024. 1, 2

[9] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19729–19739, 2023. 1, 2

[10] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *Proceedings of the*

European Conference on Computer Vision (ECCV), pages 1–18, 2022. 1, 2

[11] Mingjie Pan, Jiaming Liu, Renrui Zhang, Peixiang Huang, Xiaoqi Li, Hongwei Xie, Bing Wang, Li Liu, and Shanghang Zhang. Renderocc: Vision-centric 3d occupancy prediction with 2d rendering supervision. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12404–12411, 2024. 2

[12] Songyou Peng, Kyle Genova, Chiyu "Max" Jiang, Andrea Tagliasacchi, Marc Pollefeys, and Thomas Funkhouser. Openscene: 3d scene understanding with open vocabularies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–824, 2023. 1

[13] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 1, 2

[14] Antonin Vobecky, Oriane Siméoni, David Hurych, Spyridon Gidaris, Andrei Bursuc, Patrick Pérez, and Josef Sivic. Pop-3d: Open-vocabulary 3d occupancy prediction from images. In *Advances in Neural Information Processing Systems*, pages 50545–50557. Curran Associates, Inc., 2023. 1, 2

[15] Yi Wei, Linqing Zhao, Wenzhao Zheng, Zheng Zhu, Jie Zhou, and Jiwen Lu. Surroundocc: Multi-camera 3d occupancy prediction for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 21729–21740, 2023. 1, 2

[16] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, and Jianfeng Gao. Regionclip: Region-based language-image pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16793–16803, 2022. 2