2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

RoboEye: Enhancing 2D Robotic Object Identification with Selective 3D Geometric Keypoint Matching

Xingwu Zhang[1], Guanxuan Li[1], Zhuocheng Zhang[1] and Zijun Long[1†]

***Abstract*— The rapidly growing number of product categories in large-scale e-commerce makes accurate object identification for automated packing in warehouses substantially more difficult. As the catalog grows, intra-class variability and a long tail of rare or visually similar items increase. When combined with diverse packaging, cluttered containers, frequent occlusion, and large viewpoint changes, these factors amplify discrepancies between query and reference images, causing sharp performance drops for methods that rely solely on 2D appearance features. Thus, we propose RoboEye, a two-stage identification framework that dynamically augments 2D semantic features with domain-adapted 3D reasoning and lightweight adapters to bridge training–deployment gaps. In the first stage, we train a large vision model to extract 2D features for generating candidate rankings. A lightweight 3D-feature-awareness module then estimates 3D feature quality and predicts whether 3D re-ranking is necessary, preventing performance degradation and avoiding unnecessary computation. When invoked, the second stage uses our robot 3D retrieval transformer, comprising a 3D feature extractor that produces geometry-aware dense features and a keypoint-based matcher that computes keypoint-correspondence confidences between query and reference images instead of conventional cosine-similarity scoring. Experiments show that RoboEye improves Recall@1 by up to 7.1% over the prior state-of-the-art (RoboLLM). Moreover, RoboEye operates using only RGB images, avoiding reliance on explicit 3D inputs and reducing deployment costs. The code used in this paper is publicly available at `https://github.com/longkukuhi/RoboEye`.**

I. INTRODUCTION

Object identification (ID) aims to identify the category of a query image, which is fundamental to warehouse automation. In the pre-pick stage, correct ID within the source container provides semantic and geometric attributes (e.g., material properties and grasp points) required for motion planning [1] and control [2]; in the post-pick stage, ID governs handling and order-fulfillment accuracy. At the e-commerce scale, even small misidentification rates can produce substantial financial losses [3]. As reported in [4], Amazon's Q1 2025 included about US$1 billion in charges related to "customer returns and tariff-induced inventory adjustments."

The rapid expansion of product catalogs, coupled with the complexity of robotic warehouse environments, makes accurate ID for automated packing increasingly challenging [5], [6]. Effective image ID requires visual features that are simultaneously discriminative and robust to operational variations. The objective is to maximize similarity between query and reference images of the same class (yielding high Recall@k) while minimizing similarity across different classes. Four major factors undermine this objective: viewpoint and pose variation, occlusion and clutter, packaging and intra-class variability, and inter-class similarity combined with long-tail distributions. As catalogs scale, these factors interact to introduce more visually similar items and sparser viewpoint coverage, crowding the feature space and reducing separability.

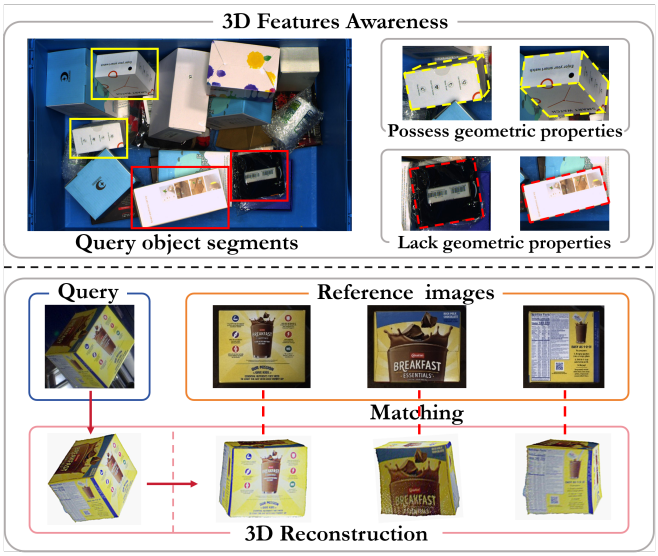

Fig. 1: Object identification under warehouse conditions. Upper part: a cluttered container with candidate items, where some query segments provide sufficient geometric cues for 3D reasoning while others lack reliable geometry. Lower part: query–reference discrepancies from viewpoint and packaging variations, where 3D geometry-aware features learned from reconstruction enables more robust verification.

The ARMBench benchmark from Amazon [6] exemplifies these conditions, showing that state-of-the-art systems (e.g., RoboLLM [7]) relying exclusively on 2D appearance cues—such as texture, color, and local gradients—are particularly vulnerable. Because these cues lack invariance to viewpoint shifts, occlusions, and packaging variations, their discriminative power sharply deteriorates under real-world warehouse settings (shown in Fig. 1) [8], [9]. Consequently, methods based solely on 2D features fail to generalize reliably across these challenging scenarios.

To address these challenges, we leverage geometry-aware image features learned for 3D reconstruction, which provide viewpoint-invariant characteristics and are inherently less sensitive to the problematic variations encountered in ware-

[1]College of Electrical and Information Engineering, Hunan University
† Corresponding author longzijun@hnu.edu.cn

house environments. A straightforward approach would be to incorporate explicit 3D inputs to complement 2D features and mitigate viewpoint-induced discrepancies. However, explicit 3D data (e.g., point clouds or depth maps) requires specialized sensors such as LiDAR or depth cameras, which increase deployment costs and complicate large-scale integration. This leads to the central question of this work:

> *How can 3D geometric cues be used to improve ID robustness under challenging warehouse conditions, without relying on explicit 3D inputs?*

To answer this, we introduce **RoboEye**, a two-stage ID framework that augments strong 2D representations with selective, domain-adapted implicit 3D reasoning and lightweight adapters to bridge training–deployment gaps. This hybrid design both reinforces appearance-based signals and supplies viewpoint-invariant geometric cues, enabling robust ID when 2D features alone are insufficient. Specifically, in the first stage (see the upper part of Fig. 2), a large vision model produces robust 2D embeddings for initial ranking. A lightweight *3D-feature-awareness* module with an MRR-driven 3D-awareness training scheme then determines whether geometric cues in the input image can be effectively exploited (demonstrated in the upper part of Fig. 1), avoiding unnecessary computation when 2D features are already discriminative and preventing performance degradation from noisy 3D cues. If 3D cues are deemed useful, the second stage invokes our proposed robot 3D retrieval transformer, which integrates a multi-view 3D feature extractor with a keypoint-based retrieval matcher. The extractor generates geometry-aware representations across views, while the matcher evaluates keypoint correspondences between query and reference images.

Our main contributions are:

- We propose *RoboEye*, the first framework to dynamically augment 2D appearance-based retrieval with domain-adapted implicit 3D geometric re-ranking, enabling robust ID without explicit 3D inputs.
- We develop an MRR-driven 3D-awareness training scheme for the *3D-feature-awareness* module that selectively activates 3D re-ranking only when beneficial.
- We introduce a 3D keypoint-based retrieval matcher that establishes confidence-weighted keypoint correspondences, offering a more robust similarity measure than conventional cosine similarity.
- We develop an adapter-based training strategy for the robot 3D retrieval transformer, enabling efficient domain adaptation.
- Our RoboEye outperforms the previous state of the art, RoboLLM, by up to 7.1% on Recall@1, as demonstrated by extensive experiments on the Amazon ARMBench dataset.

II. RELATED WORK

A. Robotic Object Identification

The object identification (ID) task in ARMBench is essentially a specialized image retrieval problem [10], where a segmented query is matched against a reference gallery. This has long been studied in computer vision with applications in robotic manipulation [11], [12], visual localization [13], and place recognition [14]. Early works [15] leveraged transformer-based classifiers with siamese matching, while later methods improved robustness via multimodal feature fusion [10] and centroid triplet loss for variable-sized inputs [16]. Most recently, RoboLLM [7] integrated the large vision model BEiT-3 [17] with contrastive training [18]–[20], achieving state-of-the-art performance with lightweight adaptations. However, most existing approaches rely mainly on 2D features without fully exploiting the 3D geometric cues inherent in observations, which are crucial for robust ID under challenging warehouse conditions such as viewpoint shifts, occlusions, and packaging variations.

B. 3D Geometric Features for Identification

Comprehending 3D geometric features is imperative for practical applications such as robotics [21], [22] and autonomous driving [23]. Current geometry-aware retrieval methods operate primarily through three interconnected paradigms: Depth-driven approaches [24], which reconstruct pseudo-point clouds from monocular RGB images but depend heavily on depth accuracy; direct 3D alignment, which leverages pre-scanned models for hierarchical feature matching [25]; and cross-modal fusion, which aligns semantics with 3D geometry via contrastive learning [26], typically requiring synthetic point clouds. In parallel, 2D-centric strategies, including local–global descriptor fusion [27] and attention-based keypoint extraction [28], remain spatially limited. Critically, most approaches rely on explicit 3D inputs—depth sensors [29], point clouds [30], or pre-rendered models [31]—introducing deployment challenges in RGB-only settings and computational bottlenecks for real-time operation.

Recent 3D foundation models such as the Visual Geometry Grounded Transformer (VGGT) [32] have demonstrated the ability to infer 3D geometry from multi-view 2D images using large-scale spatial priors. Leveraging VGGT-derived geometry, RoboEye enhances identification robustness in large-scale benchmarks while avoiding explicit 3D inputs.

III. METHOD

A. Overview

As shown in Fig. 2, the proposed *RoboEye* implements a two-stage warehouse object identification (ID) framework, dynamically augmenting 2D features with 3D cues. The first stage extracts discriminative 2D features to produce high-confidence candidates, with a lightweight 3D-feature-awareness module trained on an MRR-driven 3D-awareness objective deciding whether geometric reasoning is possible and necessary. If invoked, the second stage applies a robot 3D retrieval transformer comprising a 3D feature extractor and a keypoint-based matcher, which replaces cosine similarity with confidence-based keypoint correspondences for robust re-ranking under viewpoint shifts, occlusion, and packaging

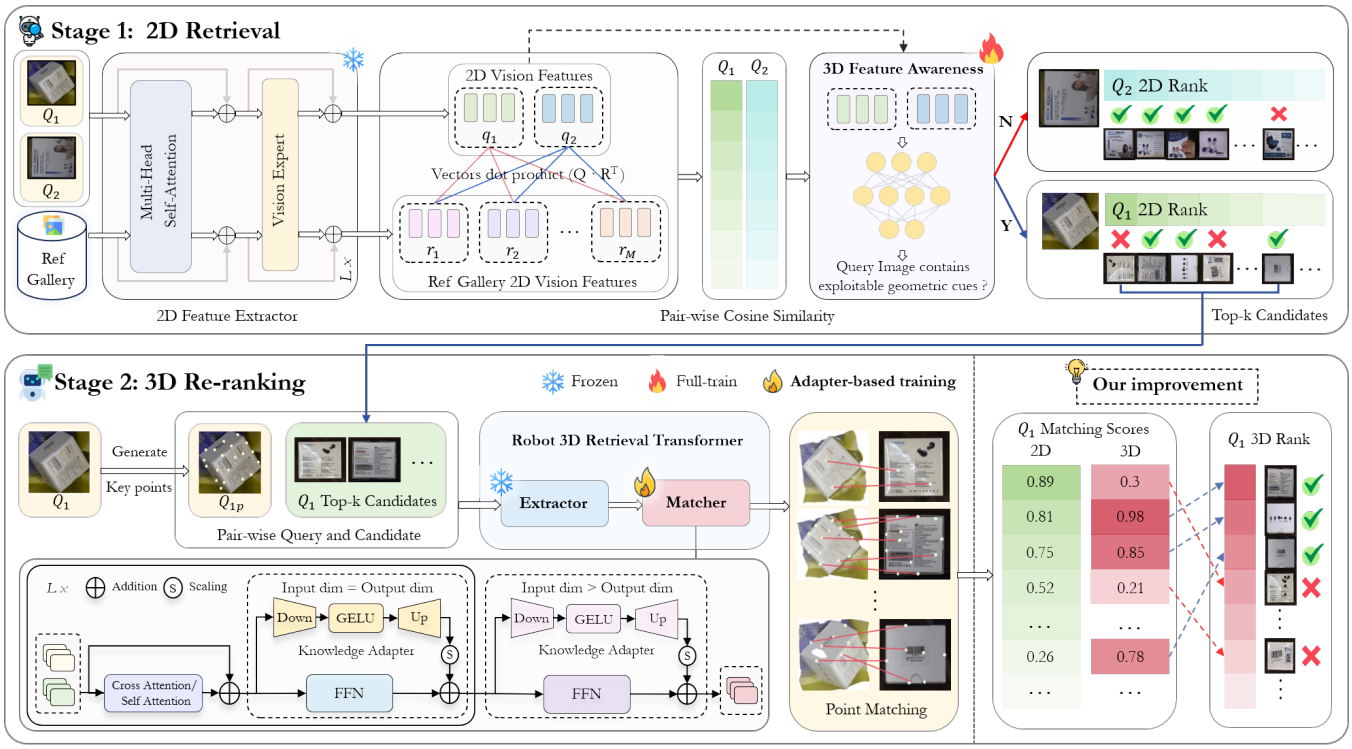

Fig. 2: Overall architecture of *RoboEye*. The framework follows a two-stage 2D→3D retrieval pipeline: (i) 2D semantic retrieval with a feature extractor and a 3D-feature-awareness module that determines the need for geometric verification; (ii) 3D re-ranking using a transformer with a feature extractor and a keypoint-based matcher to compute confidence-driven correspondences. An adapter-based training scheme ensures efficient adaptation to warehouse-specific conditions.

variations. To adapt these modules to warehouse conditions, we employ an efficient adapter-based training strategy, achieving strong performance under limited computational resources.

Subsequent sections first present the specific configurations for the object ID task, followed by a detailed description of each stage of our framework and training methods.

B. Task Definition

Given the inherent inefficiency and inaccuracy of classifying a vast number of categories with a dense layer, we formalize the object ID task in warehouse operations as an image retrieval problem, where query images consist of segmented object patches acquired via instance segmentation and the reference gallery consists of predefined catalog images, each corresponding to a known object. In practice, query images appear in two configurations: (1) single-view setting, typical of pre-pick scenarios, where the system must identify an object from the cluttered and often occluded view inside the source container; (2) multi-view setting, which arises when both pre-pick and post-pick images of the same object are available, offering complementary viewpoints that increase the likelihood of capturing reliable geometric cues. The core challenge lies in robustly matching query-gallery pairs despite significant geometric variations.

C. Stage One: 2D Retrieval

The initial stage involves an effective 2D-only feature retrieval process, where we generate a preliminary candidate ranking based on the cosine similarity of global appearance descriptors, followed by a 3D-feature-awareness module that adaptively determines the possibility and necessity of the subsequent 3D re-ranking stage.

1) 2D Feature Extractor: The crucial 2D features serve as foundational anchors for our framework, enabling rapid candidate screening while preserving essential appearance cues that guide subsequent 3D geometric re-ranking. To meet the need for robust object ID, we employ the pre-trained large model BEiT-3 [17] with an architecture featuring the multiway transformer design and strong performance across different visual tasks. Each multiway transformer block comprises a shared self-attention module and a set of modality experts, optimized for distinct input modalities. In our implementation, we streamline this architecture for visual-only inputs by retaining solely the vision-specific expert, reducing parameter overhead, as shown in the upper left of Fig. 2.

To address varying levels of detail in object ID, we consider two input scenarios: given a sequence of input images in the single-view setting $\{Q_i\}_{i=1}^{N}$, or multi-view setting $\{Q_i^0, Q_i^1, Q_i^2\}_{i=1}^{N}$, the feature extractor encodes these inputs into feature vectors. From the token dimension, we select the `[CLS]` token $\{q_i\}_{i=1}^{N} \in \mathbb{R}^{N\times d_{2D}}$ or $\{q_i^0, q_i^1, q_i^2\}_{i=1}^{N} \in \mathbb{R}^{N\times\lfloor 3\times d_{2D}\rfloor}$ as the global representation of each query, serving as a compact summary of its overall semantics. In the multi-view setting, concatenated `[CLS]` tokens proceed

through a multi-layer perceptron to map the feature dimension, providing a fused representation. All reference images $\{R_i\}_{i=1}^{M}$ undergo the same encoding pipeline, yielding `[CLS]` tokens $\{r_i\}_{i=1}^{M} \in \mathbb{R}^{M\times d_{2D}}$. The similarity between each query–reference pair is computed as the dot product between `[CLS]` tokens, yielding the initial 2D retrieval ranking.

Building on this encoding pipeline, we further adapt the extractor to the warehouse environment by training it independently based on the methodology of RoboLLM [7], using a contrastive loss to align the features for matched query–reference pairs.

2) 3D-feature-awareness Module: Although 2D-only ranking demonstrates decent performance, challenging warehouse ID cases still require complementary 3D geometric cues, as explained in the introduction section. In exploring how to exploit 3D features, we identified three key issues. First, certain queries inherently lack sufficient geometric cues, as demonstrated in the upper part of Fig. 1. Second, indiscriminate 3D re-ranking can degrade accuracy because mismatches between 2D and 3D signals introduce noise that outweighs geometric benefits (see Section V-B). Third, 3D re-ranking incurs substantial computational overhead.

To address these issues, we propose a 3D-feature-awareness module that, given the initial 2D features, (1) assesses whether reliable 3D features can be extracted and (2) decides if 3D re-ranking is warranted. By skipping unnecessary 3D processing, the module reduces the second-stage inference cost and strikes a balance between accuracy and efficiency. Further details on the architectures and the training methodology of this module are presented in the following section.

3) MRR-driven 3D-awareness Training: We train the 3D-feature-awareness module to cooperate with the 2D feature extractor using MRR-driven 3D-awareness training (M3AT). The module is deliberately kept lightweight, consisting of only a few dense layers. While its size could be increased to achieve higher performance, doing so would also introduce greater computational cost and inference time. Its novelty and effectiveness arise from M3AT, a supervision scheme that identifies when 3D re-ranking provides tangible benefits and trains the module to detect implicit geometric cues within 2D features.

Concretely, for each query we compute the Mean Reciprocal Rank (MRR) of the initial 2D ranking and the MRR after applying 3D re-ranking. A query is labeled positive if its MRR improves after 3D re-ranking, and negative otherwise. To address the label imbalance, we optimize a class-weighted cross-entropy loss with class-specific weights w, assigning a larger weight to positive examples.

The module maps a 2D feature $q_i \in \mathbb{R}^{1\times d_{2D}}$ to class logits via a two-layer projection:

$$h_i = \mathrm{GELU}\left(\mathrm{LN}(q_i)\mathbf{W}_{\mathrm{hidden}}\right), \tag{1}$$

$$\tilde{q}_i = h_i\mathbf{W}_{\mathrm{class}} \in \mathbb{R}^2, \tag{2}$$

where $\mathbf{W}_{\mathrm{hidden}} \in \mathbb{R}^{d_{2D}\times\tilde{d}}$ and $\mathbf{W}_{\mathrm{class}} \in \mathbb{R}^{\tilde{d}\times 2}$. Here LN denotes LayerNorm. The training objective is the weighted cross-entropy:

$$\mathcal{L}_{\mathrm{M3AT}} = \mathrm{CE}_w\left(\mathrm{softmax}(\tilde{q}_i),\, y_i\right),$$

where y_i is the MRR-derived binary label and CE_w applies weights w to the classes.

At inference, the module predicts whether to enable 3D re-ranking. If a query is predicted positive, we select the top-K reference candidates from the initial 2D retrieval ranking and refine them using the 3D pipeline; otherwise the initial 2D ranking is returned. M3AT therefore concentrates the contribution on the training signal—learning when geometric reasoning is truly beneficial—while keeping the awareness module computationally trivial.

D. Stage Two: 3D Geometric Re-ranking

The lack of invariance in 2D appearance cues under warehouse settings amplifies discrepancies between query and reference images (shown in the lower part of Fig. 1). To address this, we introduce a robot 3D retrieval transformer, consisting of a 3D feature extractor and a 3D keypoint-based retrieval matcher (illustrated in the lower-left part of Fig. 2). The extractor encodes 3D cues into compact geometry-aware representations, while the matcher establishes keypoint correspondences between query and reference images, replacing conventional cosine similarity with a more robust confidence-based measure. Crucially, this module operates directly on 2D query-reference image pairs, avoiding the need for explicit 3D inputs such as point clouds or depth maps and augmenting 2D representations with a robust geometric verification.

1) 3D Feature Extractor: We take the aggregator component in VGGT [32] as the 3D feature extractor in our framework. It employs a transformer architecture with alternating frame-wise and global self-attention layers. This design captures both intra-view spatial structure and cross-view geometric relations, yielding compact 3D-aware feature representations. These representations form the foundation for our proposed 3D point-based matching method.

2) 3D Keypoint-based Retrieval Matcher: Once 3D dense features are extracted, the key challenge is how to leverage them effectively for retrieval. We observe that directly applying cosine similarity to 3D features provides poor discrimination (see Section V-B). We argue that this is because global 3D representations introduce additional noise for matching, whereas local geometric correspondences—conditioned on viewpoint and pose—are more discriminative than global fine-grained similarity. To address this limitation, we aim to design a 3D keypoint-based retrieval matcher that replaces cosine similarity with correspondence-driven scoring, providing more reliable compensation for 2D appearance-based ranking.

The track head of VGGT [32] partially meets our purpose and integrates well with the 3D feature extractor by leveraging large-scale end-to-end pre-training. The matcher is a transformer that processes a grid of point-match tokens, each initialized with appearance features and enriched with correlation cues for geometric alignment. Tokens encode

position, visibility, and confidence, and are iteratively refined through interleaved self- and cross-attention, yielding correspondences across views.

However, the original track head was designed for point tracking, focusing primarily on finding correspondences across images. In contrast, our task requires a more sophisticated similarity measurement between query–reference pairs to support re-ranking and category ID. To this end, we adapt its mechanism to not only establish correspondences but also generate confidence scores as similarity estimates. We redesign it as a retrieval matcher that evaluates geometric consistency between candidate pairs, aggregating these scores into a ranking criterion that replaces conventional cosine similarity and yields more robust results.

Therefore, direct reuse of the track head in the VGGT model is ineffective due to both the shift in task objectives and the distribution gap between warehouse-specific datasets and VGGT's pre-training data. To address the change in matcher's functionality, we introduce a 3D keypoint-based matching mechanism. To mitigate the distribution gap while ensuring practical deployment, we further develop an adapter-based domain adaptation strategy. The following sections provide detailed descriptions of these components.

3D Keypoint-based Matching Mechanism. The top-K query–reference image pairs $\{Q_i, R_i\}_{i=1}^{K}$ obtained from the 2D semantic ranking serve as input to the re-ranking stage.

The proposed matching mechanism is based on sparse keypoint matching rather than whole-image comparison. In the following, we describe the procedure for a single query–reference pair, which is applied independently to all candidates. The 3D feature extractor first processes the image pair through alternating frame-wise and global self-attention layers, producing a set of 3D tokens T^{3D}. Subsequently, the matcher takes the 3D tokens T^{3D} to predict geometric tracking features F_q, F_r. Keypoints $(x_j, y_j)_{j=1}^{S}$ are then detected in each query image Q_i using SIFT [33], where S denotes the number of keypoints. For each keypoint, its feature representation is extracted by bilinear sampling from the query feature map F_q and then correlated with the feature map of reference candidate F_r. These correlation maps are processed through attention layers to predict matched keypoints $(\hat{x}_j, \hat{y}_j)_{j=1}^{S}$ together with confidence scores $\{C_j\}_{j=1}^{S}$. Formally, the final similarity score $\tilde{C}$ for a candidate is computed as the mean confidence over all keypoints:

$$\tilde{C} = \frac{1}{S} \sum_{j=1}^{S} \{C_j\}_{j=1}^{S} \tag{3}$$

In the multi-view scenario, each query view is paired with the same reference candidate and processed as above, and the final score is obtained by averaging the per-view scores. The sorted final scores represent the rank of the 3D geometric similarity between each query image and reference candidate pairs.

Adapter-based Domain Shifting. To bridge the domain gap between VGGT pre-training and warehouse operation conditions, the most direct approach would be to fully re-train VGGT with our proposed matching mechanism. However, this is computationally prohibitive, as the original setup requires 64 A100 GPUs for nine days [32]. To provide a practical alternative, we design an adapter-based training strategy: rather than updating the entire network, we freeze the 3D feature extractor and confine training to the matcher, enhanced with lightweight knowledge adapters [34].

For the training objective, we construct samples by selecting the highest-ranked correct candidate for each query as a positive and the three top-ranked incorrect candidates as negatives. The matcher is optimized with a cross-entropy loss, which maximizes the scores of positives while suppressing those of negatives, enhancing the discriminative power of matcher for warehouse-specific scenarios.

As for the architecture, we use an adapter module to replace the feedforward network in a transformer block with a dual-path structure: a primary branch that retains the original pre-trained network, and a parallel, trainable branch dedicated to domain-specific adaptation. For parameter efficiency, this new branch employs a bottleneck architecture, featuring a down-projection and an up-projection layer. The transformation performed by the adaptive branch on an input feature x_i' to produce the new feature is formally expressed as:

$$\tilde{x_i} = \mathrm{GELU}(\mathrm{LN}(x_i') \cdot \mathbf{W}_{\text{down}}) \cdot \mathbf{W}_{\text{up}} \tag{4}$$

In this formulation, $\mathbf{W}_{\text{down}} \in \mathbb{R}^{d \times \tilde{d}}$ and $\mathbf{W}_{\text{up}} \in \mathbb{R}^{\tilde{d} \times d'}$ represent the down-projection and up-projection layers. LN stands for LayerNorm. d' can be set to match either the input dimension d or the final output dimension of the transformer block and $\tilde{d}$ corresponds to the latent dimension of the bottleneck, satisfying $\tilde{d} \ll d$. The output of this bottleneck module is integrated with the original FFN path via a scaled residual connection modulated by a factor α. Subsequently, the combined features from both paths, $\tilde{x_i}$ and x_i', are fused with the initial input x_i through a final residual connection:

$$x_i = FFN\left(LN\left(x_i'\right)\right) + \alpha \cdot \tilde{x}_i + x_i' \tag{5}$$

IV. EXPERIMENTS

A. Experimental Setup

Dataset and Configuration. The experiments use the ARMBench dataset [6] provided by Amazon, a large-scale benchmark containing over 190,000 unique items and reflecting realistic warehouse conditions. We consider two query settings: (1) single-view: one pre-pick image per item; (2) multi-view: the pre-pick image augmented by two post-pick images. We evaluate under two reference-gallery scenarios: (a) container gallery, where entries correspond to containers holding multiple objects; and (b) global gallery, containing all unique objects across the dataset.

Training Details. The results reported for the BEiT-3 Large model are based on a training methodology consistent with that of the 2D feature extractor in RoboEye. Both the 2D feature extractor and the corresponding components of the robot 3D retrieval transformer are initialized with their pre-trained weights, while knowledge adapters and the

Model	Ref Set	Recall@1		Recall@2		Recall@3	
		N=1	N=3	N=1	N=3	N=1	N=3
ResNet50-RMAC	Container	71.7	72.2	81.9	82.9	87.2	88.2
DINO-ViT-S	Container	77.2	79.5	87.3	89.4	91.6	93.5
BEiT-3-Base*	Container	83.7	84.5	83.8	N/A	84.5	N/A
BEiT-3-Large	Container	96.9	99.0	97.0	99.1	97.2	99.2
RoboLLM	Container	97.8	98.0	97.9	98.1	98.0	98.2
RoboEye	Container	**98.2**	**99.4**	**98.3**	**99.5**	**98.4**	**99.6**
BEiT-3-Large	Global	62.4	35.1	69.3	46.7	72.4	53.3
RoboLLM	Global	74.6	78.2	82.6	85.7	85.3	89.1
RoboEye	Global	**79.4**	**85.3**	**84.4**	**91.1**	**86.3**	**93.5**

TABLE I: Results on the ID task at varying Recall@k. N=1 denotes the single-view setting, while N=3 denotes the multi-view setting. * denotes no ARMBench training. "Ref Set" indicates whether the reference images are drawn from the container gallery or the global gallery.

3D-feature-awareness module are randomly initialized. The hidden dimension of the 3D-feature-awareness module is set to 64, equal to the hidden dimension of the knowledge adapter. The factor α of the knowledge adapter is set to 0.6.

Following previous works [6], [7], [16], performance is measured with Recall@k. The cross-entropy loss for MRR-driven 3D-awareness training uses class-specific weights w with a 4:1 ratio. During object identification (ID) experiments, we set the number of candidates for re-ranking to 16 and sample 20 keypoints.

B. Object Identification in Single-view Setting

We begin our evaluation with the single-view setting, the most common yet constrained condition. Due to the large-scale catalog and the presence of viewpoint shifts, occlusions, and packaging variations, the proposed framework must correctly identify objects under restricted viewpoint coverage and with limited 3D geometric information.

As summarized in Table I, RoboEye consistently outperforms previous state-of-the-art methods. In the most challenging case—the global gallery setting—RoboEye achieves a 4.8% improvement at Recall@1. Under the container gallery setting, RoboEye raises Recall@1 from 97.8% to 98.2% (+0.4%) compared to RoboLLM, despite the strong, near-saturated baseline. Note that the larger 2D baseline, BEiT-3 Large, suffers a sharp performance drop to only 62.4% on Recall@1 as the catalog expands, whereas RoboEye remains robust with 79.4%. This suggests that merely scaling up model size does not necessarily improve performance for this task. In contrast, RoboEye effectively counteracts degradation from viewpoint shifts, occlusions, packaging variations, and large-scale catalog growth, delivering consistent gains even with minimal inputs.

C. Object Identification in Multi-view Setting

This section evaluates the performance of RoboEye in a multi-view setting, where the framework leverages cross-view feature fusion and 3D geometric cues to enhance consistency and accuracy in object ID.

The quantitative comparison is shown in Table I. RoboEye achieves larger absolute gains in this setting compared to the single-view setting, as the additional inputs reinforce its core mechanism. When retrieving from the container gallery, Recall@1 improves from 98.0% to 99.4% (+1.4%) compared to RoboLLM. In the global gallery retrieval scenario, Recall@1 of RoboEye increases by 7.1%, establishing a clear margin over the best prior result. These findings demonstrate that RoboEye generalizes beyond single-query conditions and becomes more effective with multiple queries, further highlighting its robustness in practical warehouse environments. The framework not only maintains consistency across views but also fully exploits multi-view redundancy through feature fusion and cross-image 3D geometric reasoning.

2D-FE	3D-FE	3D-KMNT	3D-KMT	3D-FAM	Recall@1
✓	×	×	×	×	70.5
×	✓	×	×	×	2.0
✓	✓	×	×	×	18.2
✓	✓	✓	×	×	64.4
✓	✓	×	✓	×	68.2
✓	✓	×	✓	✓	**79.4**

TABLE II: Ablation results of RoboEye's components.

V. Analysis

A. Analysis Configuration

Unless otherwise stated, all analyses are conducted under the single-view setting with retrieval from the global gallery, as this represents the most common and challenging scenario. The 2D feature extractor (2D-FE) is implemented in two variants: BEiT-3 Base (2D-FE-B) and BEiT-3 Large (2D-FE-L), with the latter used only in comparison experiments to separate the effects of increased parameters from the architectural contributions of RoboEye.

B. Naive 3D Fusion vs. Selective Geometric Re-ranking

In this section, we evaluate the contributions of RoboEye's key components and outline the exploration process that led to its design. We first experimented with using only 3D geometric features, but this approach achieves poor results. We then tried a naive two-stage pipeline, using 2D features for initial retrieval and 3D features for re-ranking; however, this still underperformed compared to using 2D features alone. These findings motivated us to develop a dedicated matching mechanism and training strategy for 3D geometric retrieval, rather than relying solely on cosine similarity for measuring feature similarity. This progression ultimately resulted in the RoboEye framework proposed in this paper.

Using the 2D-FE alone, we achieve a Recall@1 of 70.5% (first line in Table II). Although we use the same backbone and training procedure as RoboLLM [7], RoboLLM reports 74.6% under this setting. We attribute this gap to hardware limitations (smaller batch sizes), since contrastive learning typically benefits from larger batches. Despite our 2D model being 4.1% weaker initially, our full framework (RoboEye) still outperforms RoboLLM by 4.8%, demonstrating its effectiveness.

Next, we evaluate the performance using only the 3D feature extractor (3D-FE). This experiment yields a very

2D-FE-B	2D-FE-L	3D-RT	3D-FAM	Time	Recall@1
✓	×	×	×	**0.028**	70.5
✓	×	✓	×	0.547	64.4
✓	×	✓	✓	0.071	**76.8**
×	✓	×	×	0.068	62.4

TABLE III: Average per-sample inference time (seconds) under different configurations.

Model	Trained Parameters	Recall@1	Recall@2	Recall@3
2D-FE-B	221.0M	70.5	78.2	81.7
2D-FE-L	672.7M	62.4	69.3	72.4
RoboEye	222.3M	**79.4**	**84.4**	**86.3**

TABLE IV: Comparison of retrieval performance and trained parameter scales between RoboEye and 2D-FE of different model sizes.

low Recall@1 of 2.0% (second line in Table II), indicating that directly using cosine similarity on 3D features as the similarity measurement is insufficient for retrieval. This poor performance motivates designing a specialized matching mechanism and a matcher model to better leverage 3D features.

We also tried a naive two-stage pipeline, in which 2D features perform the initial ranking followed by 3D feature re-ranking. However, this simple fusion only marginally improves performance: Recall@1 rises from 2.0% to 18.2% (third line in Table II), which remains far below the 2D-only baseline of 70.5%. This suggests that straightforward 3D-based re-ranking introduces significant noise, undermining the discriminative power of the 2D features.

To fully exploit multi-view 3D information, we evaluate the 3D keypoint-based retrieval matcher with two variants. The first has no warehouse-specific adaptation (3D-KMNT); the second includes adapter-based training on warehouse data (3D-KMT). Using 3D-KMNT raises Recall@1 to 64.4% (fourth line in Table II), yet this is still below the 2D-only baseline, highlighting a domain gap in applying generic geometric reasoning to warehouse images. With warehouse-adapted training (3D-KMT), Recall@1 improves to 68.2% (+3.8%, fifth line in Table II), confirming that lightweight domain adaptation helps. Still, this is below the 2D baseline, indicating that geometry alone cannot match the state-of-the-art ID performance.

Finally, we introduce a 3D-feature-awareness module (3D-FAM) to dynamically combine 3D and 2D features. This module assesses whether reliable 3D information can be extracted from a given query and whether it should be applied. By selectively enabling 3D-based re-ranking only when beneficial, we avoid performance degradation from noisy 3D data and unnecessary computation. As shown in the last line of Table II, adding the 3D-FAM yields the best result, increasing Recall@1 from 68.2% to 79.4% (+11.2%). Integrating all components together achieves the highest performance, demonstrating that domain-adapted 3D reasoning with selective activation is critical for accurate and efficient large-scale warehouse identification (ID).

C. How Does RoboEye Reduce Inference Latency?

To understand the trade-off between recognition accuracy and computational efficiency, we measure the average per-sample inference latency and Recall@1 across different configurations, as summarized in Table III. Results are reported with candidate pool size set to 4, measured on a single NVIDIA 5090 GPU with 32GB memory. The first baseline uses only the 2D-FE-B and achieves the fastest runtime with moderate accuracy, providing efficient candidate retrieval but without 3D geometric verification. Incorporating the robot 3D retrieval transformer (3D-RT) without any awareness control (see second line in Table III) increases latency from 0.028 to 0.547 seconds while even reducing Recall@1 by 6.1%, reflecting the computational burden of unconditional 3D re-ranking without performance gain. With the 3D-FAM (third line in Table III), Recall@1 improves by 6.3% while latency remains low at 0.071 seconds—nearly an order of magnitude faster than naive 3D re-ranking. The latency is also comparable to the large 2D-FE (0.068 seconds, last line in Table III), yet achieves 14.4% higher Recall@1 and retains the advantages of geometric verification.

These results highlight the pivotal role of the 3D-FAM in balancing efficiency and robustness. By activating 3D reasoning only when necessary, it preserves near-2D runtime while retaining the benefits of selective geometric verification, making the approach practical for large-scale warehouse deployments where both latency and reliability are critical.

D. Does a Larger Model Always Help?

In this section, we investigate whether simply increasing the size of 2D-FE can obviate the need for exploiting 3D geometric features. To this end, we perform a controlled comparison between the full RoboEye pipeline and two variants of the 2D-FE with different parameter sizes.

As shown in Table IV, scaling the 2D-FE from 221.0M to 672.7M parameters not only increases training cost but also leads to degraded performance, with Recall@1 dropping by 8.1%. In contrast, RoboEye achieves an improvement of 17.0% in Recall@1 over the 2D-FE-L while using only 222.3M trained parameters—essentially the same scale as 2D-FE-B and merely one-third the size of 2D-FE-L—yet delivering a clear performance gain.

This comparison shows that enlarging the 2D backbone alone is insufficient to overcome warehouse-specific challenges such as occlusion, clutter, packaging variation, and inter-class visual similarity. By contrast, RoboEye leverages complementary geometric reasoning, enabled by its robot 3D retrieval transformer and 3D awareness-guided re-ranking, to provide robust improvements at a comparable parameter scale. These results underscore the necessity of incorporating 3D-aware components rather than relying solely on larger 2D models for effective large-scale warehouse ID.

E. Impact of 2D Candidate Pool Size

In this section, we evaluate the impact of the candidate-pool size K, introduced by RoboEye's two-stage ranking

Model	Candidates	Recall@1	Recall@2	Recall@3
RoboEye	4	76.8	81.0	82.7
RoboEye	8	78.5	83.1	84.9
RoboEye	**16**	**79.4**	**84.4**	**86.3**
RoboEye	32	78.8	83.8	85.8
RoboEye	64	78.5	83.3	85.3

TABLE V: ID performance of RoboEye with varying candidate pool sizes (K) after applying the full two-stage pipeline.

design, on final performance.

We first evaluate the recall of the 2D-FE-B alone at varying cutoff thresholds K. This measurement defines the upper bound of stage two performance, as 3D re-ranking is based on the 2D ranking result and any candidate not retrieved in stage one cannot be recovered by subsequent re-ranking. Specifically, recall rises from 84.7% at a pool size of 4 to 87.4%, 90.5%, 92.6%, and 94.3% as the pool expands to 8, 16, 32, and 64, respectively. Expanding the pool size from 4 to 16 yields a 5.8% gain, whereas enlarging the pool from 16 to 64 (a fourfold increase) brings only a modest 3.8% improvement. While larger candidate pools provide slightly higher coverage of the ground-truth object, they also impose substantially greater computational cost during 3D re-ranking. Conversely, smaller pools risk excluding the correct object, thereby capping overall performance.

Beyond the upper-bound analysis, we further investigate how different candidate pool sizes affect the end-to-end performance of RoboEye. The results in Table V reveal several important trends. Increasing the pool size from 4 to 16 raises Recall@1 by 2.6%, yielding the best overall results. However, further enlarging the pool to 32 or 64 candidates does not yield additional gains; instead, performance slightly declines (Recall@1 drops by 0.6% and 0.9%, respectively). This counterintuitive trend suggests that including too many low-quality candidates introduces noise into the geometric verification stage, which can dilute discriminative signals and destabilize re-ranking. Most importantly, these results indicate that RoboEye is insensitive to the candidate-pool size K: values from 8 to 64 yield comparable performance. Taken together, we adopt $K = 16$ as the default configuration, which strikes a balance between retrieval coverage, re-ranking effectiveness, and computational cost.

VI. CONCLUSIONS

We presented RoboEye, a two-stage identification framework that augments 2D appearance features with selective 3D reasoning. By dynamically invoking 3D re-ranking only when beneficial, RoboEye achieves robust performance without explicit 3D inputs. Experiments on the Amazon ARMBench dataset show up to 7.1% Recall@1 improvement over the previous state-of-the-art, demonstrating that RoboEye effectively mitigates the challenges posed by large-scale catalogs, viewpoint and pose variations, occlusions, and packaging appearance changes, thereby establishing RoboEye as an efficient and scalable solution for reliable object identification in warehouse automation.

REFERENCES

[1] W. Tang, J.-H. Pan *et al.*, "Embodiment-agnostic action planning via object-part scene flow," in *ICRA*, 2025.
[2] S. Noh, J. Kim *et al.*, "Graspsam: When segment anything model meets grasp detection," in *ICRA*, 2025.
[3] E. Aduh, F. Wang *et al.*, "Avoiding object damage in robotic manipulation," in *IROS*, 2024.
[4] E. Kim, "Amazon took a mysterious $1 billion hit from customer returns and tariff maneuvering," https://www.businessinsider.com/amazon-billion-hit-customer-returns-tariffs-2025-5, 2025, business Insider.
[5] N. Correll, K. E. Bekris *et al.*, "Analysis and observations from the first amazon picking challenge," *T-ASE*, 2016.
[6] C. Mitash, F. Wang *et al.*, "Armbench: An object-centric benchmark dataset for robotic manipulation," *arXiv preprint:2303.16382*, 2023.
[7] Z. Long, G. Killick *et al.*, "Robollm: Robotic vision tasks grounded on multimodal large language models," in *ICRA*, 2024.
[8] C. Eppner, S. Höfer *et al.*, "Lessons from the amazon picking challenge: Four aspects of building robotic systems." in *RSS*, 2016.
[9] S. Back, S. Lee *et al.*, "High-quality unknown object instance segmentation via quadruple boundary error refinement," in *ICRA*, 2025.
[10] C. Mitash, M. Hussein *et al.*, "Scaling object-centric robotic manipulation with multimodal object identification," in *ICRA*, 2024.
[11] A. Zeng, S. Song *et al.*, "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," *IJRR*, 2022.
[12] N. Di Palo and E. Johns, "Dinobot: Robot manipulation via retrieval and alignment with vision foundation models," in *ICRA*, 2024.
[13] A. Anoosheh, T. Sattler *et al.*, "Night-to-day image translation for retrieval-based localization," in *ICRA*, 2019.
[14] Z. Chen, F. Maffra *et al.*, "Only look once, mining distinctive landmarks from convnet for visual place recognition," in *IROS*, 2017.
[15] A. Gouda and M. Roidl, "Dounseen: Tuning-free class-adaptive object detection of unseen objects for robotic grasping," *arXiv preprint:2304.02833*, 2023.
[16] A. Gouda *et al.*, "Learning embeddings with centroid triplet loss for object identification in robotic grasping," in *CASE*, 2024.
[17] W. Wang, H. Bao *et al.*, "Image as a foreign language: Beit pretraining for vision and vision-language tasks," in *CVPR*, 2023.
[18] Z. Zhang, K. Liang *et al.*, "Eliminating hallucination in diffusion-augmented interactive text-to-image retrieval," *arXiv preprint arXiv:2601.20391*, 2026.
[19] Z. Long, L. Zhuang *et al.*, "Understanding and mitigating human-labelling errors in supervised contrastive learning," in *ECCV*, 2024.
[20] Z. Long *et al.*, "Multiway-adapter: Adapting multimodal large language models for scalable image-text retrieval," in *ICASSP*, 2024.
[21] X. Yan, J. Hsu *et al.*, "Learning 6-dof grasping interaction via deep geometry-aware 3d representations," in *ICRA*, 2018.
[22] Y. Li, Y. Zhang *et al.*, "Representing robot geometry as distance fields: Applications to whole-body manipulation," in *ICRA*, 2024.
[23] W. Sun, X. Lin *et al.*, "Sparsedrive: End-to-end autonomous driving via sparse scene representation," in *ICRA*, 2025.
[24] T. Yang, Y. Ju *et al.*, "Imov3d: Learning open vocabulary point clouds 3d object detection from only 2d images," *NeurIPS*, 2024.
[25] H. Zhou, A.-A. Liu *et al.*, "Dual-level embedding alignment network for 2d image-based 3d object retrieval," in *ACM MM*, 2019.
[26] D. Hegde, J. M. J. Valanarasu *et al.*, "Clip goes 3d: Leveraging prompt tuning for language grounded 3d recognition," in *ICCV*, 2023.
[27] M. Yang, D. He *et al.*, "Dolg: Single-stage image retrieval with deep orthogonal fusion of local and global features," in *ICCV*, 2021.
[28] P.-E. Sarlin, A. Unagar *et al.*, "Back to the feature: Learning robust camera localization from pixels to pose," in *CVPR*, 2021.
[29] M. Danielczuk, A. Kurenkov *et al.*, "Mechanical search: Multi-step retrieval of a target object occluded by clutter," in *ICRA*, 2019.
[30] L. Wiesmann, R. Marcuzzi *et al.*, "Retriever: Point cloud retrieval in compressed 3d maps," in *ICRA*, 2022.
[31] P. Ausserlechner, D. Haberger *et al.*, "Zs6d: Zero-shot 6d object pose estimation using vision transformers," in *ICRA*, 2024.
[32] J. Wang, M. Chen *et al.*, "Vggt: Visual geometry grounded transformer," in *CVPR*, 2025.
[33] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, 2004.
[34] N. Houlsby, A. Giurgiu *et al.*, "Parameter-efficient transfer learning for nlp," in *ICML*, 2019.