

Distilling 3D distinctive local descriptors for 6D pose estimation

Amir Hamza*,†, Andrea Caraffa*, Davide Boscaini*, Fabio Poiesi*

*Fondazione Bruno Kessler, †University of Trento

Abstract—Three-dimensional local descriptors are crucial for encoding geometric surface properties, making them essential for various point cloud understanding tasks. Among these descriptors, GeDi has demonstrated strong zero-shot 6D pose estimation capabilities but remains computationally impractical for real-world applications due to its expensive inference process. *Can we retain GeDi’s effectiveness while significantly improving its efficiency?* In this paper, we explore this question by introducing a knowledge distillation framework that trains an efficient student model to regress local descriptors from a GeDi teacher. Our key contributions include: an efficient large-scale training procedure that ensures robustness to occlusions and partial observations while operating under compute and storage constraints, and a novel loss formulation that handles weak supervision from non-distinctive teacher descriptors. We validate our approach on five BOP Benchmark datasets and demonstrate a significant reduction in inference time while maintaining competitive performance with existing methods, bringing zero-shot 6D pose estimation closer to real-time feasibility.

I. INTRODUCTION

Three-dimensional descriptors (or features) encode geometric surface properties through compact numerical representations [1]. They are essential for various downstream tasks in point cloud understanding, including registration [2], object 6D pose estimation [3], segmentation [4], [5], and navigation [6]. These descriptors can be broadly categorized into *local* and *global* based on the information they encode. Local descriptors operate at the point or patch level, capturing fine-grained geometric details [2], [7]–[9]. Global descriptors also provide point-level representations, but aggregate geometric information from the entire point cloud [10]–[12]. Global descriptors are generally more computationally efficient [13] but often sacrifice generalization across different data domains [2].

Local descriptors can achieve rotation invariance through local reference frames (LRF) [2] or point-pair features (PPF) [14]. LRF-based methods establish a local coordinate system for each point using surrounding geometric information [15]. PPF-based methods compute antisymmetric 4D features from pairs of 3D points and their surface normals [16]. Among local descriptors, GeDi [2] has been successfully applied to 6D pose estimation¹ through the FreeZe method [3] in a zero-shot setting, achieving performances that, for the

This work was supported by the Horizon Europe projects AI-PRISM (Grant Agreement 101058589) and FORTIS (Grant Agreement 101135707). We acknowledge ISCRA for awarding this project access to the LEONARDO supercomputer, owned by the EuroHPC Joint Undertaking, hosted by CINECA (Italy).

¹6D pose estimation involves determining the rotation and translation of a query object (e.g., CAD model) relative to the reference frame of a target object in a scene (e.g., captured by an RGB-D camera).

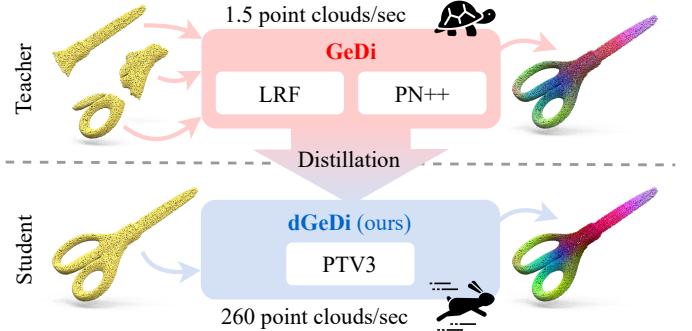


Fig. 1. We introduce dGeDi, a 3D point cloud encoder trained by distilling GeDi [2] features (Teacher, Top) into a PointTransformerV3 [18] (PTV3) backbone (Student, Bottom). GeDi suffers from slow inference as it processes points sequentially (multiple input arrows), first extracting local reference frames (LRF) and then computing descriptors with PointNet++ [17] (PN++). Instead, dGeDi retains GeDi’s generalization and distinctiveness while being over 170 times faster, making it ideal for real-time robotics applications.

first time in the literature, nearly match those of state-of-the-art fully-supervised methods. However, GeDi’s inference time is impractical for real-world applications due to its computationally expensive process: LRF calculation [15] for each point followed by a PointNet++-based descriptor encoding step [17] (top Fig. 1).

In this work, we explore how to address this inefficiency by using knowledge distillation [19] to train a student model (bottom Fig. 1) that regresses GeDi local descriptors, i.e., the teacher model. However, this presents several challenges. Distillation in this context requires large-scale point cloud datasets [20]–[22]. The rotation and scale invariance properties of the teacher model have to be effectively transferred to the student model. Low inlier ratios must be handled, as only a small subset of descriptors is typically reliable, resulting in a significant imbalance during distillation, where most points provide unreliable supervision. Despite these challenges, our approach presents unique opportunities. A more efficient descriptor extractor could bring zero-shot 6D pose estimation closer to real-time performance, making it more viable for practical applications. Moreover, by tailoring the distillation process specifically for 6D pose estimation, we can improve descriptor robustness to occlusions and partial observations. To the best of our knowledge, this is the first attempt to apply distillation to 3D local descriptors.

To this end, we introduce a distillation approach to train an efficient neural network (student) based on PointTransformerV3 (PTV3) [18] to generate GeDi descriptors (teacher). We refer to our approach as dGeDi, derived from distilled GeDi. A key challenge when distilling GeDi is balancing

efficiency and practicality. On the one hand, a naive online distillation approach, where the teacher is computed jointly with the student, is impractical due to GeDi’s slow run-time, making training excessively slow. On the other hand, a naive offline distillation, where teacher features are precomputed, would be highly storage-inefficient. To overcome these limitations, we propose a novel *learning via correspondences* paradigm. Offline, we precompute features from a query object, while online, we transfer them to the target object present in the scene using ground-truth transformations from synthetic 6D pose estimation datasets. This allows scalable training on large datasets while keeping compute and storage requirements manageable. Another challenge is that, unlike regression, where supervision comes from precise ground-truth annotations, our distillation supervision can include non-distinctive descriptors (e.g., on the BOP Benchmark [23], only 3% of points can be matched through nearest neighbor search in the teacher feature space). Standard loss functions [24], [25] tend to overfit these outliers, producing unreliable student features. To mitigate this, we introduce a novel loss function that weights each point’s contribution based on its registration error. Points with errors above a certain threshold contribute almost equally, preventing excessive penalization from outliers. Since training data are synthetic, we incorporate data augmentation to enhance rotational robustness in the student model and reduce the synthetic-to-real domain gap. We validate dGeDi through direct comparison with GeDi [2] and further evaluate its effectiveness by integrating it into FreeZe [3], a state-of-the-art zero-shot pose estimator. Additionally, we conduct an extensive ablation study to analyze dGeDi’s key components. Experimental results demonstrate that dGeDi significantly reduces inference time while maintaining competitive performance with alternative methods. In summary, our main contributions are:

- We introduce a distillation approach to learn local 3D descriptors for zero-shot 6D pose estimation using an efficient neural network, achieving a 170-times speed-up without compromising accuracy.
- We propose a data-scaling technique that enables training on large-scale synthetic data while significantly reducing memory requirements.
- We design a novel loss function that weights the regression error to prevent overfitting on synthetic data, thereby improving generalization.

II. RELATED WORKS

A. Zero-shot object 6D pose estimation

Zero-shot pose estimation methods estimate the 6D pose of objects that are unseen during training. These methods train on synthetic dataset or leverage the knowledge acquired by pre-trained vision and geometric foundational models. The first category of methods like MegaPose [26] and ZeroPose [27] uses render and compare strategy. At first a classification network selects the best-matching template from a set of rendered images that closely resemble the input image. Then, a refiner network iteratively determines the pose by matching

the selected template with additional viewpoints. Since these method requires a separate inference for each input image and template pair, the computational cost and inference time scale linearly with the number of templates. Methods like GigaPose [28] addresses the issue of inference time scaling. It renders templates to extract dense features using a Vision Transformer (ViT), and then finds the best template using fast nearest neighbour search in the feature space. On top, two lightweight MLPs estimate the 2D scale and in plane rotation from a single 2D-2D correspondence and the two relative patch features extracted by another feature extractor. SAM-6D [29] leverages ViT to extract features that are back-projected into 3D to find 3D-3D correspondences between two point clouds. SAM-6D uses a matching network to enhance features before finding the actual correspondences. FoundationPose [30] propose a generalizable and unified pipeline for pose tracking. The coarse pose is estimated using a trained transformer based network. This is followed by a pose refinement network that operates on input image crops and the renderings of the object model. The refined poses are then ranked using a self attention based transformer network.

Other methods do not require task-specific training and leverage the knowledge learned by large scale vision and geometric foundation models. ZS6D [31] extracts features using pre-trained DINOv2 from the inputs image and templates. Then, the best template is chosen using the cosine similarity and 2D-3D correspondences are retrieved using patch-wise similarity. The pose is estimated using PnP and RANSAC. FoundPose [32] integrates DINOv2 patch features into bag-of-words representation for efficient template retrieval. The pose hypothesis is generated using PnP and RANSAC and is refined using optimization-based photometric alignment algorithm. FreeZe [3] fuses descriptors from pre-trained geometric and vision foundation models to extract discriminative features. This is followed by RANSAC based registration and a symmetry-aware pose refinement.

In this work, we use FreeZe’s geometric encoder as a teacher to train a more efficient student encoder via distillation. This approach preserves the discriminative power of the original model while significantly improving efficiency.

B. Feature distillation

Knowledge distillation involves training a compact model using information from a more complex model or an ensemble of features [33]. While this concept has been explored in various computer vision tasks, such as object detection [34]–[36], image recognition [37], [38], and segmentation [39], [40], its application to 6D pose estimation has been less explored. Guo et al. [41] introduce knowledge distillation for 6D pose estimation by supervising the student with the teacher’s local predictions. Both teacher and student networks process an RGB image to generate segmentation masks and local predictions, which are then used as correspondences for a PnP solver. HRPose [42] distills features from the teacher network using an efficient neural network architecture. The student network is trained with supervision from both ground-truth 6D poses and geometric features via a feature

similarity loss. CleanPose [43] integrates causal learning with residual-based knowledge distillation to transfer rich category information from a pre-trained 3D encoder based on ULIPI-2. The frozen 3D encoder extracts a global point cloud descriptor, while the pose estimation network computes an average-pooled descriptor.

In this work, we distil knowledge from descriptors computed by a frozen teacher model, without relying on intermediate representations. Instead, our student is trained to regress features, which are then used for pose computation via feature matching. Like CleanPose, we use a frozen 3D encoder as the teacher, but instead of global CLIP-aligned representations, our approach uses rotation-invariant, highly distinctive point-level features from a local 3D encoder.

C. Geometric encoders

Geometric encoders map 3D points to high-dimensional feature representations that capture local geometric information. Existing methods operate at either *point* or *voxel* level. *Point-based* approaches process raw point clouds directly, either by encoding points independently [17], [44], [45] or within local neighborhoods to incorporate contextual information [9], [46]–[50]. *Voxel-based* approaches first quantize point clouds into voxel grids at a given resolution and then efficiently process non-empty voxels using sparse 3D convolutions [13], [51] or space-filling curves [18]. While voxelization improves scalability, it comes at the cost of fine geometric details.

In this work, we use a point-based encoder [2] as the teacher, which provides distinctive descriptor representations but incurs slower inference. Meanwhile, we employ a voxel-based encoder [18] as the student, enabling faster runtime. The teacher relies on a custom module to estimate local reference frames (LRF in Fig. 1), ensuring rotation invariance by design. In contrast, the student follows a simple feed-forward architecture, requiring no custom modules, and achieves robustness through knowledge distillation.

III. PRELIMINARIES: FREEZE METHOD

Among existing zero-shot pose estimation methods in the literature [26], [29], [30], we base our work on FreeZe [3], which achieved state-of-the-art accuracy, winning the BOP Challenge 2024 [52]. FreeZe has three main stages: preprocessing, feature extraction, and feature matching.

Problem formulation. Given an RGBD image $\mathbf{I} \in \mathbb{R}^{H \times W \times 4}$ capturing a scene and the 3D model \mathcal{O} of an object within the scene, 6D pose estimation aims to determine the rigid transformation \mathbf{T} of \mathcal{O} relative to the reference frame of the sensor that captured \mathbf{I} . Let Q be the *query* object, and T be the *target* object representing \mathcal{O} 's partial observation within \mathbf{I} . We define $\mathbf{T}^{Q \rightarrow T} = (\mathbf{R}, \mathbf{t})$, where $\mathbf{R} \in SO(3)$ is the rotation and $\mathbf{t} \in \mathbb{R}^3$ is translation of Q .

Preprocessing. Let \mathcal{P}^Q and \mathcal{P}^T be the 3D point cloud of Q and T , respectively. \mathcal{P}^Q is obtained by sampling points on \mathcal{O} 's surface. \mathcal{P}^T is obtained by using an off-the-shelf zero-shot segmentor [29] to extract the crop $\mathbf{I}^T \subset \mathbf{I}$ containing T , and then lifting \mathbf{I}^T in 3D using sensor's intrinsic parameters.

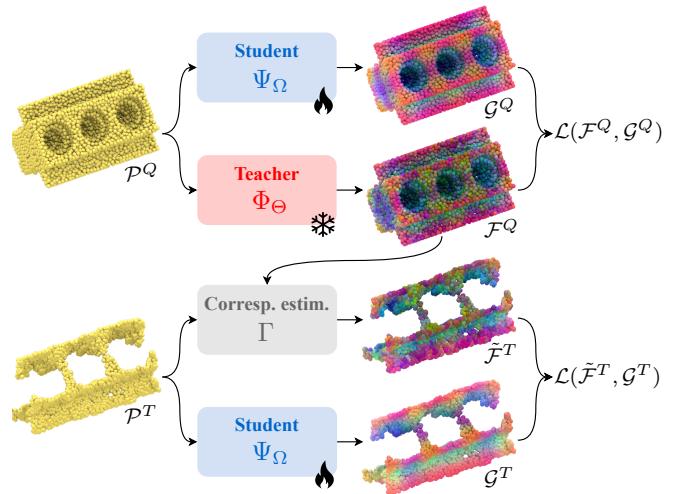


Fig. 2. Overview of dGeDi. Top: A Query point cloud is being fed to both teacher and student network. We obtain distinctive 3D local descriptor from (frozen) teacher which guide the student descriptors via proposed distillation loss. Bottom: Rather than computing the teacher features for Target objects, we leverage the learning via correspondences. We transfer the features from query to target point cloud using the ground-truth transformation available at training time. By leveraging this, we significantly reduce the memory and compute overhead.

Feature extraction. FreeZe extracts features \mathcal{F}^Q , \mathcal{F}^T from \mathcal{P}^Q , \mathcal{P}^T using frozen vision [53] and geometric [2] encoders, i.e., $\mathcal{F} = (\mathcal{F}_{\text{geo}} | \mathcal{F}_{\text{vis}})$, where \mathcal{F}_{geo} are the geometric features, \mathcal{F}_{vis} are the visual features, and $|$ is the concatenation operator.

Feature matching. The transformation $\mathbf{T}^{Q \rightarrow T}$ is estimated using RANSAC [54] and refined with ICP [55]. Point-level correspondences between \mathcal{P}^Q and \mathcal{P}^T are established via nearest neighbor search in the feature space. Next, triplets of correspondences are sampled, and those leading to inconsistent alignments are discarded. For each valid triplet, the transformation is computed using Singular Value Decomposition (SVD), and the hypothesis with the highest inliers ratio is selected. Symmetries are solved using SAR [3].

IV. OBJECT-ORIENTED DISTILLATION

While FreeZe performs feature extraction with both geometric and vision encoders, we focus on the distillation of former ones², assuming $\mathcal{F} = \mathcal{F}_{\text{geo}}$. We introduce a novel distillation procedure to transfer the distinctive properties of a slow teacher encoder to a faster student encoder (Fig. 2). We define our proposed distillation approach as *object-oriented* because training supervision is provided at the object level. First, we extract teacher descriptors $\mathcal{F}^Q = \Phi_{\Theta}(\mathcal{P}^Q)$, $\mathcal{F}^T = \Phi_{\Theta}(\mathcal{P}^T)$ using GeDi [2] encoder Φ_{Θ} . Then, we learn student descriptors $\mathcal{G}^Q = \Psi_{\Omega}(\mathcal{P}^Q)$, $\mathcal{G}^T = \Psi_{\Omega}(\mathcal{P}^T)$ with a PTV3 [18] encoder Ψ_{Ω} . During distillation, we optimize the parameters Ω so that $\mathcal{G}^Q \approx \mathcal{F}^Q$ and $\mathcal{G}^T \approx \mathcal{F}^T$, while Θ remain frozen. Note that this differs from online knowledge distillation approaches [56], where both the teacher and student are neural networks and learn Θ , Ω simultaneously. In our case, the architectures of Φ_{Θ} and Ψ_{Ω} are distinct (Fig. 1). Teacher

²Object-oriented distillation can also be applied to the vision encoder, but it is out of scope for this work.

features are precomputed, stored in memory, and loaded as needed during distillation. To optimize this process, we propose storing teacher features only for query objects and introduce a module that leverages ground-truth 6D poses to transfer them to target objects (Sec. IV-A). Moreover, we propose a custom loss function that focuses learning on noise-free points, leading to improved performance (Sec. IV-B).

A. Learning via correspondences

During training, given a data sample $(\mathcal{P}^Q, \mathcal{P}^T, \mathbf{T}^{Q \rightarrow T})$, we leverage the ground-truth 6D pose $\mathbf{T}^{Q \rightarrow T}$ to align \mathcal{P}^Q with \mathcal{P}^T and perform a nearest neighbor search in the metric space to establish the correspondence $\Gamma: \mathcal{P}^T \rightarrow \mathcal{P}^Q$, where each point $\mathbf{p} \in \mathcal{P}^T$ is mapped to $\mathbf{q} = \operatorname{argmin}_{\mathbf{q} \in \mathcal{P}^Q} \|\mathbf{T}^{Q \rightarrow T} \mathbf{q} - \mathbf{p}\|$, where \mathbf{p} and \mathbf{q} represent the 3D coordinates of corresponding points. We use Γ to transfer query teacher descriptors $\mathcal{F}^Q = \Phi_\Theta(\mathcal{P}^Q)$ to \mathcal{P}^T , obtaining $\tilde{\mathcal{F}}^T = \Gamma(\mathcal{F}^Q)$. By replacing the explicit computation of $\mathcal{F}^T = \Phi_\Theta(\mathcal{P}^T)$ with $\tilde{\mathcal{F}}^T$ during distillation, we achieve two key benefits: (i) improved supervision for target objects and (ii) a scalable, object-oriented distillation process.

Improved target supervision. Matching features $\mathcal{F}^Q, \mathcal{F}^T$ is challenging because input data differ significantly: \mathcal{P}^Q is computed from noise-free synthetic CAD models, while \mathcal{P}^T is extracted from noisy real-world RGBD images. This domain gap results in variations in point densities, partiality, and artifacts created by lifting the input image \mathbf{I} to generate \mathcal{P}^T . For example, on the MegaPose-GSO [26] dataset, we measured that, on average, only 21.4% of the points can be matched via nearest neighbor search in the teacher feature space. This means that 78.6% of the features are considered outliers for feature matching. In contrast, by transferring query teacher features to target point clouds with Γ , we simulate the ideal scenario where all target points are matched with the corresponding query points, promoting reliable supervision during distillation.

Improved scalability by reducing storage requirements. The MegaPose-GSO dataset consists of 850 query objects and 16M target objects, resulting in a 1:19k ratio. This ratio can be further expanded by applying random transformations to the existing data. As a result, storing target teacher features for a large-scale training dataset becomes prohibitively expensive in both computation and storage. For example, the entire MegaPose [26] dataset would require 14TB of storage. In contrast, our approach significantly improves scalability by storing only the query teacher features, reducing the storage requirements for MegaPose by 411 times (i.e., 34GB).

B. Loss function design

Although GeDi has proven to be both generalizable and distinctive [2], it still exhibits low inlier ratios, with an average RON of 3% on the BOP Benchmark and 21.4% on MegaPose-GSO. This indicates that only a small fraction of points are highly distinctive, while the majority have ambiguous features. Consequently, a loss function that assigns equal weight to all points will be dominated by outliers, leading to overfitting on

ambiguous features. To solve this issue, we propose a novel loss function:

$$\mathcal{L}(\mathcal{F}, \mathcal{G}) = \begin{cases} \alpha \|\mathcal{F} - \mathcal{G}\|^2, & \text{if } \|\mathcal{F} - \mathcal{G}\| < \delta \\ \beta \|\mathcal{F} - \mathcal{G}\| + \gamma, & \text{otherwise.} \end{cases} \quad (1)$$

Eq. 1 is inspired by Huber loss [24], with a quadratic behavior for points with low regression error (below a threshold δ) and a linear behavior otherwise. In our experiments, we set $\alpha = 8$, $\beta = 1/5$, $\gamma = 8\delta^2 - \delta/5$, and $\delta = 0.3$. With these hyperparameters, the quadratic component is relatively steep, while the linear component is nearly flat. This design achieves two key objectives: (i) for low regression errors, the significant variation in loss values encourages accurate regression; (ii) for high regression errors, the minimal variation in loss values prevents excessive focus on outliers, assigning significantly smaller loss value compared to traditional loss designs such as MSE and Huber losses. To validate this assumption, in Sec. V-B we present an experimental comparison with an alternative loss formulation that prioritizes learning from points with high regression error:

$$\tilde{\mathcal{L}}(\mathcal{F}, \mathcal{G}) = \|\mathcal{F} - \mathcal{G}\|^\epsilon \mathcal{L}(\mathcal{F}, \mathcal{G}), \quad (2)$$

where \mathcal{L} is from Eq. 1 and ϵ is a hyper-parameter inspired by the Focal loss [25]. In our experiments, we set $\epsilon = 15$.

V. EXPERIMENTS

A. Experimental setup

We validate dGeDi with three experiments: (i) a direct comparison against GeDi [2], (ii) a detailed analysis of dGeDi's key components, and (iii) a comparison with state-of-the-art zero-shot pose estimators by integrating dGeDi in FreeZe [3]. During preprocessing, we randomly sample 4k points from \mathcal{P}^Q and \mathcal{P}^T , and apply statistical outlier removal [57] to \mathcal{P}^T to reduce depth noise. Both point clouds are normalized by \mathcal{P}^Q 's diameter to promote scale invariance. In (i) and (ii), we perform feature extraction using only the geometric encoder, while in (iii) we integrate dGeDi with FreeZe's vision encoder. In (i) and (iii), we use both ground-truth and SAM-6D [29] segmentation masks to ensure fairness and compliance with the zero-shot assumption. In (ii), we perform object localization using ground-truth segmentation masks to focus the analysis on feature quality without localization errors. GeDi extracts 32-dimensional point-level features from local neighborhoods covering 30% of Q 's diameter. After feature extraction, we perform feature matching using the Open3D's RANSAC and ICP implementations. We run 100k iterations for RANSAC and 1k iterations for ICP. The student model is trained using eight NVIDIA A100 GPUs, while evaluation is performed on a single NVIDIA A40 GPU. Note that the peak memory usage of dGeDi when processing a point cloud with 5k points is only 717MB, making it suitable for deployment on edge devices with limited resources.

Training dataset. We train dGeDi on the MegaPose-ShapeNetCore [26] dataset, which contains 1 million synthetic images generated by rendering various 3D objects against

random background scenes. The dataset provides ground-truth segmentation masks, which we use to localize target objects.

Evaluation datasets. We validate dGeDi on the BOP Benchmark datasets with publicly available ground-truth 6D poses: LM-O [58], T-LESS [59], TUD-L [60], IC-BIN [61], and YCB-V [62]. This benchmark presents several challenges, comprising a diverse set of everyday and industrial objects, both textured and texture-less objects, symmetries, varying levels of occlusion, and different lighting conditions.

Evaluation metrics. We evaluate the accuracy of dGeDi using the ratio-of-nearest points (RON) [6] and feature-matching recall (FMR) [14] metrics. RON measures the percentage of query points with a nearest neighbor in the target object within a given distance threshold τ_1 . FMR measures the percentage of points with RON greater than τ_2 . During validation, we set $\tau_1 = 3\%$ of Q 's diameters and $\tau_2 = 5\%$. Note that both metrics do not require registration, providing a direct assessment of feature quality. To evaluate the 6D pose estimation accuracy, we use average recall (AR) [23], as in the BOP Benchmark. We evaluate the efficiency of dGeDi by measuring the average processing time per image in seconds.

B. Quantitative results

GeDi vs. dGeDi. Tab. I compares GeDi [2] (teacher, Rows 1-12) and dGeDi (student, Rows 13-24) on the BOP Benchmark using different types of segmentation masks. When using ground-truth segmentation masks, dGeDi significantly outperforms GeDi by +4.1 RON, +11.8 FMR, and +2.4 AR, while being 170 times faster (Row 25). dGeDi outperforms GeDi in RON and FMR scores across all datasets but shows a slight drop in AR on TUD-L [60] and YCB-V [62]. This difference arises because RON and FMR directly assess feature quality, whereas AR is computed post registration, where RANSAC's robustness to outliers can obscure the true contribution of the features. On T-LESS [59], dGeDi achieves a substantial +14.1 AR improvement over the teacher. We attribute this significant gain to the specific type of 3D models: unlike other datasets, T-LESS provides CAD models of industrial objects with fine-grained geometric details and sharp edges. As shown in Fig. 3, dGeDi excels in this setting, producing features that are smoother than the teacher's and better capture the underlying geometry. When using SAM-6D zero-shot segmentation masks, dGeDi outperforms the teacher by +2.7 RON and +9.4 FMR, but achieves a -1.1 AR (Row 26). This setting is more challenging than the former because zero-shot segmentation masks may have inaccuracies along object contours, can miss some parts of the object, and can include parts of other objects or background, while during distillation we use ground-truth segmentation masks. Despite this, dGeDi proves to be robust to the domain gap, achieving a superior feature quality compared to the teacher (better RON and FMR) and a comparable AR. Beyond improved accuracy, dGeDi also achieves superior computational efficiency, extracting features at more than 50 frames per second. Given that each image contains an average of five target objects, this implies that dGeDi can process approximately 260 point clouds per

TABLE I

COMPARISON OF GEDI AND DGEIDI ON THE BOP BENCHMARK IN TERMS OF FEATURE QUALITY (RON, FMR) AND POSE ESTIMATION ACCURACY (AR) WITH GROUND-TRUTH (GT) AND ZERO-SHOT (ZS) SEGMENTATION MASKS. AVERAGE FEATURE EXTRACTION TIME PER IMAGE IS REPORTED IN SECONDS. ROWS 25-26 HIGHLIGHT DGEIDI IMPROVEMENT OVER GEDI.

Dataset	GT	ZS	RON ↑	FMR ↑	AR ↑	Run-time ↓
GeDi [2]						
1 LM-O	✓	✓	1.8	8.8	66.9	3.198
2			3.5	7.4	60.7	
dGeDi (ours)						
13 LM-O	✓	✓	4.3	27.2	67.2	0.021
14			1.7	23.5	60.4	
15 T-LESS	✓	✓	2.1	15.6	58.6	0.021
16			1.6	11.6	31.6	
17 TUD-L	✓	✓	16.3	77.8	93.2	0.003
18			12.8	67.3	83.0	
19 IC-BIN	✓	✓	3.7	18.5	33.1	0.034
20			4.5	2.6	31.3	
21 YCB-V	✓	✓	9.1	42.3	58.8	0.015
22			7.4	34.8	52.3	
23 Avg	✓	✓	7.1	36.3	62.2	0.019
24			5.6	27.9	51.7	
25 Δ	✓	✓	+4.1 +2.7	+11.8 +9.4	+2.4 -1.1	173×
26						

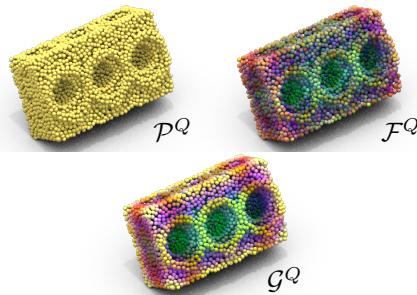


Fig. 3. Comparison between teacher (center) and student (right) features on a query object from T-LESS (left). Colors represent PCA-reduced features. Student features \mathcal{F}^Q are smoother and less noisy than teacher features \mathcal{P}^Q .

second. Processing times (rightmost column) depends on the number of object instances per image, ranging from 3ms for a single instance in TUD-L to 34ms for dozens of instances in IC-BIN [61]. In summary, dGeDi delivers high accuracy with low latency, making it suitable for time-critical robotics applications.

Ablation study. Tab. II analyses the impact of three key components in dGeDi: supervision type (Col. 1), loss function design (Cols. 2-5), and data augmentation strategy (Col. 6-8). Row 1 is our baseline, which uses supervision only from

query objects, Eq. 1 as loss, and no data augmentation. Row 9 is dGeDi’s final configuration, while Rows 2-8 explore alternative designs. We evaluate each variant on the BOP Benchmark and report its improvement over the baseline in the rightmost column. We draw three main insights for these experiments. First, the learning via correspondence paradigm introduced in Sec. IV-A improves performance considerably (+5.2 AR, Row 5) by introducing supervision from partial and noisy target objects with the same memory requirements. Second, the proposed loss (Eq. 1) outperforms both traditional designs (+3.9 AR against MSE in Row 2, +3.5 AR against Huber in Row 3) and Eq. 2 (+8.7 AR, Row 4). This indicates that effective distillation of frozen GeDi features requires a careful loss design, and that reducing the contribution of points with a large regression error is beneficial. Third, data augmentation promote generalization (Row 9 achieves +0.4 AR over Row 5). These ablation experiments confirm that integrating correspondence-based supervision, a tailored loss, and data augmentation consistently enhances performance.

dGeDi and FreeZe integration. Tab. III assesses the 6D pose estimation performance obtained when integrating dGeDi in FreeZe. We compare against state-of-the-art zero-shot 6D pose estimators on the same datasets as Tab. I, evaluating AR and run-time. We compare the performance of dGeDi with SAM-6D [29], FoundationPose [30], and the original version of FreeZe, which are the state-of-the-art zero-shot methods. Rows 1-3 present publicly available results from the BOP Benchmark leaderboard [63]. Rows 4-5 present the reproduced results of FreeZe, both with and without symmetry-aware pose refinement (SAR). Rows 6-7 show performance when replacing FreeZe’s geometric encoder GeDi [2] with dGeDi. To ensure a fair comparison, all methods use the SAM-6D zero-shot segmentor for object localization except for Rows 8-9 where ground-truth segmentation masks are used. For experiments using ground-truth segmentation masks, dGeDi achieves the same AR of FreeZe, while reducing the inference time by 2.7 seconds (Row 9). When using SAM-6D segmentation masks, dGeDi reaches 70.4 AR compared to FreeZe’s 71.5, yet it runs 5.2 seconds faster. This run-time improvement is mainly due to dGeDi ability to process multiple target objects per image in parallel, unlike FreeZe, which processes them serially. In contrast, the slight performance drop is mainly due to noise in the predicted segmentation masks. Since dGeDi is trained with ground-truth, noise-free segmentation masks, the introduction of noisy SAM-6D segmentation masks creates a domain gap that leads to suboptimal pose estimation, although dGeDi still maintains better feature quality and faster processing.

C. Qualitative results

Figure 4 presents qualitative results with a clear layout: the first column shows the input images, the second displays predictions from dGeDi, and the third contains those from GeDi. For clarity, the predicted poses are overlaid on the input images and the RON is indicated for each case. Each row corresponds to an example drawn from a different dataset, addressing various challenges such as occlusions (a, d, e),

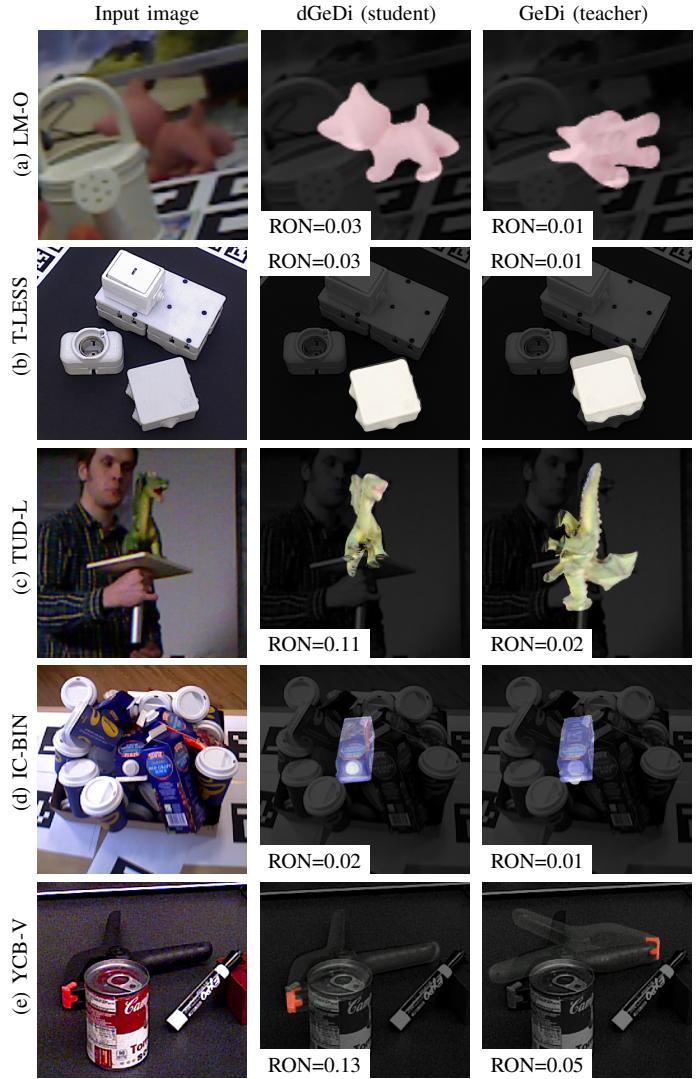


Fig. 4. Qualitative results on the BOP Benchmark (left) comparing dGeDi (center) with GeDi (right). We overlay the object’s 3D model transformed according to the predicted pose on the greyscale input image for better contrast. Key challenges: occlusions (a, d, e), object symmetry (b), partial view (c), multiple instances (d).

object symmetries (b), partial views (c), and the presence of multiple instances of the same object type (d). In examples (a) and (e), the toy cat is partially obscured by a watering can (from LM-O) and clipper tool (from YCBV) is obscured by a food can. GeDi fails in these cases due to the insufficient visible structure needed to reliably form accurate LRFs. This leads to incorrect pose predictions. In contrast, dGeDi excels by directly distilling robust local geometric descriptors from the available cues. This targeted learning helps dGeDi to accurately recover the pose even when only limited geometric information is present. In (d), a juice box is heavily occluded by cups in a challenging bin picking scenario involving multiple instances. Although the teacher’s prediction (with a RON of 0.01) appears quantitatively favorable, its reliance on sparse LRF cues leads to unreliable pose estimates under these adverse occlusions. In this case, dGeDi achieves a RON of 0.02 by extracting and utilizing richer local geometric details, thereby successfully recovering the correct pose.

TABLE II

ABLATION ANALYSIS ON THREE KEY COMPONENTS: SUPERVISION STRATEGY, LOSS FUNCTION, AND DATA AUGMENTATION. THE RIGHTMOST COLUMN INDICATES EACH VARIANT’S IMPROVEMENT OVER THE BASELINE (ROW 1). ROW 9 SHOWS THE FINAL CONFIGURATION OF DGEDI.

Target obj. supervision	Loss function			Data augmentation			BOP Benchmark								
	MSE	Huber	Eq. 2	Eq. 1	Rotation	Jitter	Point drop.	LM-O	T-LESS	TUD-L	IC-BIN	YCB-V	Avg	Δ	
1				✓				64.5	47.9	88.5	30.8	51.3	56.6	-	
2	✓	✓						59.5	54.9	86.4	32.2	56.7	57.9	+1.3	
3	✓		✓					61.3	56.2	85.5	31.5	56.8	58.3	+1.7	
4	✓			✓				51.2	49.0	77.2	32.1	56.1	53.1	-3.5	
5	✓				✓			67.5	56.8	94.1	32.8	57.8	61.8	+5.2	
6	✓				✓	✓		68.9	56.5	94.2	32.0	58.0	61.9	+5.3	
7	✓					✓		67.8	56.8	93.6	33.0	58.2	61.9	+5.3	
8	✓						✓	67.7	57.9	93.5	32.8	58.5	62.1	+5.5	
9	✓			✓	✓	✓	✓	67.2	58.6	93.2	33.1	58.8	62.2	+5.6	

TABLE III

RESULTS ON THE BOP BENCHMARK IN TERMS OF AR AND RUN-TIME WITH GROUND-TRUTH (GT) AND ZERO-SHOT (ZS) SEGMENTATION MASKS.

Method	Segm.		AR \uparrow						Run-time (s) \downarrow		
	GT	ZS	LM-O	T-LESS	TUD-L	IC-BIN	YCB-V	Avg	Avg	GPU	
1 SAM-6D [29]		✓	69.9	51.5	90.4	58.8	84.5	71.0	4.5	RTX3090	
2 FoundationPose [30]		✓	75.6	64.6	92.3	50.8	88.9	74.4	35.9	A100	
3 FreeZe [3]		✓	71.6	53.1	94.9	54.5	84.0	71.6	12.7	A40	
4 FreeZe [3] w/o SAR		✓	71.4	53.0	94.8	51.5	83.5	70.9	10.8	A40	
5 FreeZe [3]		✓	71.4	53.0	94.8	54.3	83.9	71.5	13.7	A40	
6 FreeZe w/ dGeDi w/o SAR (ours)		✓	69.6	53.5	93.1	49.1	79.7	69.0	6.4	A40	
7 FreeZe w/ dGeDi (ours)		✓	69.6	53.5	93.1	52.3	83.4	70.4	8.5	A40	
8 FreeZe [3] w/o SAR	✓		80.2	75.1	98.6	53.8	90.2	79.6	8.7	A40	
9 FreeZe w/ dGeDi w/o SAR (ours)	✓		80.5	75.0	98.7	54.2	89.4	79.6	6.0	A40	

despite the limited and cluttered geometric information. In (b), a symmetrical square electrical component is the target. dGeDi learns fine discriminative features even in the presence of symmetry, overcoming the ambiguities that affect the GeDi’s LRF. By focusing on fine patterns and slight asymmetries within a geometrically symmetrical structure, dGeDi achieves an accurate pose prediction. In summary, for all cases, dGeDi generates more distinctive descriptors compared to GeDi (higher RON) and successfully recovers the correct 6D poses.

VI. CONCLUSIONS

In this work, we explored knowledge distillation for 3D local descriptors, aiming to retain the effectiveness of GeDi while significantly improving its efficiency. By training an efficient student model to regress GeDi descriptors, we addressed key challenges such as the need for large-scale point cloud datasets, the transfer of rotation and scale invariance, and handling low inlier ratios during distillation. Our approach explicitly enforces consistency between query and target descriptors, enhancing robustness to occlusions and partial observations. Experimental results demonstrate that our distilled model achieves a substantial reduction in inference time while maintaining competitive performance with alternative methods. This brings zero-shot 6D pose estimation closer to real-time feasibility, making it more viable for practical applications. To the best of our knowledge, this is the first attempt to apply knowledge distillation to 3D local

descriptors, opening new possibilities for efficient geometric reasoning in real-world scenarios.

Limitations. Our work focuses solely on improving the efficiency of geometric feature extraction within the FreeZe pipeline. While this enhances performance, other components, such as registration, remain computational bottlenecks.

Future works. We plan to explore strategies to accelerate these remaining modules, further optimizing the entire 6D pose estimation pipeline for real-time applications.

REFERENCES

- [1] R. Rostami, F. S. Bashiri, B. Rostami, and Z. Yu, “A survey on data-driven 3D shape descriptors,” *CGF*, 2019.
- [2] F. Poiesi and D. Boscaini, “Learning general and distinctive 3D local deep descriptors for point cloud registration,” *TPAMI*, 2023.
- [3] A. Caraffa, D. Boscaini, A. Hamza, and F. Poiesi, “FreeZe: Training-free zero-shot 6D pose estimation with geometric and vision foundation models,” in *ECCV*, 2024.
- [4] C. Saltori, E. Krivosheev, S. Lathuilière, N. Sebe, F. Galasso, G. Fiameni, E. Ricci, and F. Poiesi, “GIPSO: Geometrically Informed Propagation for Online Adaptation in 3D LiDAR Segmentation,” in *ECCV*, 2022.
- [5] M. Garosi, R. Tedoldi, D. Boscaini, M. Mancini, N. Sebe, and F. Poiesi, “3D Part Segmentation via Geometric Aggregation of 2D Visual Features,” in *WACV*, 2025.
- [6] Y. Zhou, Y. Wang, F. Poiesi, Q. Qin, and Y. Wan, “Loop closure detection using local 3D deep descriptors,” *RAL*, 2022.
- [7] S. Ao, Q. Hu, B. Yang, A. Markham, and Y. Guo, “SpinNet: Learning a general surface descriptor for 3D point cloud registration,” in *CVPR*, 2021.
- [8] H. Wang, Y. Liu, Z. Dong, and W. Wang, “You only hypothesize once: Point cloud registration with rotation-equivariant descriptors,” in *ACM MM*, 2022.

- [9] D. Boscaini and F. Poiesi, “PatchMixer: Rethinking network design to boost generalization for 3D point cloud understanding,” *IVC*, 2023.
- [10] H. Deng, T. Birdal, and S. Ilic, “Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors,” in *ECCV*, 2018.
- [11] Z. J. Yew and G. H. Lee, “Regrt: End-to-end point cloud correspondences with transformers,” in *CVPR*, 2022.
- [12] H. Yu, J. Hou, Z. Qin, M. Saleh, I. Shugurov, K. Wang, B. Busam, and S. Ilic, “RIGA: Rotation-invariant and globally-aware descriptors for point cloud registration,” *TPAMI*, 2024.
- [13] C. Choy, J. Park, and V. Koltun, “Fully Convolutional Geometric Features,” in *ICCV*, 2019.
- [14] H. Deng, T. Birdal, and S. Ilic, “Ppfnet: Global context aware local features for robust 3d point matching,” in *CVPR*, 2018.
- [15] J. Yang, Q. Zhang, Y. Xiao, and Z. Cao, “Toldi: An effective and robust approach for 3d local shape description,” *Pattern Recognition*, 2017.
- [16] B. Drost, M. Ulrich, N. Navab, and S. Ilic, “Model globally, match locally: Efficient and robust 3D object recognition,” in *CVPR*, 2010.
- [17] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, “PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space,” *NeurIPS*, 2017.
- [18] X. Wu, L. Jiang, P.-S. Wang, Z. Liu, X. Liu, Y. Qiao, W. Ouyang, T. He, and H. Zhao, “Point Transformer V3: Simpler Faster Stronger,” in *CVPR*, 2024.
- [19] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” in *NeurIPS*, 2014.
- [20] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al., “Shapenet: An information-rich 3d model repository,” *arXiv:1512.03012*, 2015.
- [21] L. Downs, A. Francis, N. Koenig, B. Kinman, R. Hickman, K. Reymann, T. B. McHugh, and V. Vanhoucke, “Google scanned objects: A high-quality dataset of 3d scanned household items,” in *ICRA*, 2022.
- [22] M. Deitke, R. Liu, M. Wallingford, H. Ngo, O. Michel, A. Kusupati, A. Fan, C. Laforte, V. Voleti, S. Y. Gadre, et al., “Objaverse-xl: A universe of 10m+ 3d objects,” *NeurIPS*, 2024.
- [23] T. Hodaň, M. Sundermeyer, B. Drost, Y. Labb  , E. Brachmann, F. Michel, C. Rother, and J. Matas, “Bop challenge 2020 on 6d object localization,” in *ECCV*. Springer, 2020.
- [24] P. J. Huber, “Robust estimation of a location parameter,” *Annals of Mathematical Statistics*, 1964.
- [25] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Doll  , “Focal loss for dense object detection,” in *ICCV*, 2017.
- [26] Y. Labb  , L. Manuelli, A. Mousavian, S. Tyree, S. Birchfield, J. Tremblay, J. Carpentier, M. Aubry, D. Fox, and J. Sivic, “Megapose: 6d pose estimation of novel objects via render & compare,” in *CoRL*, 2023.
- [27] J. Chen, M. Sun, T. Bao, R. Zhao, L. Wu, and Z. He, “3d model-based zero-shot pose estimation pipeline,” *arXiv:2305.17934*, 2023.
- [28] V. N. Nguyen, T. Groueix, M. Salzmann, and V. Lepetit, “GigaPose: Fast and robust novel object pose estimation via one correspondence,” in *CVPR*, 2024.
- [29] J. Lin, L. Liu, D. Lu, and K. Jia, “SAM-6D: Segment anything model meets zero-shot 6d object pose estimation,” in *CVPR*, 2024.
- [30] B. Wen, W. Yang, J. Kautz, and S. Birchfield, “FoundationPose: Unified 6d pose estimation and tracking of novel objects,” in *CVPR*, 2024.
- [31] P. Ausserlechner, D. Haberger, S. Thalhammer, J.-B. Weibel, and M. Vincze, “ZS6D: Zero-shot 6d object pose estimation using vision transformers,” in *ICRA*, 2024.
- [32] E. P.   rnek, Y. Labb  , B. Tekin, L. Ma, C. Keskin, C. Forster, and T. Hodan, “FoundPose: Unseen object pose estimation with foundation features,” in *ECCV*, 2024.
- [33] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” *arXiv:1503.02531*, 2015.
- [34] G. Chen, W. Choi, X. Yu, T. Han, and M. Chandraker, “Learning efficient object detection models with knowledge distillation,” *NeurIPS*, 2017.
- [35] A. Chawla, H. Yin, P. Molchanov, and J. Alvarez, “Data-free knowledge distillation for object detection,” in *WACV*, 2021.
- [36] Z. Kang, P. Zhang, X. Zhang, J. Sun, and N. Zheng, “Instance-conditional knowledge distillation for object detection,” *NeurIPS*, 2021.
- [37] Z. Huang, S. Yang, M. Zhou, Z. Li, Z. Gong, and Y. Chen, “Feature map distillation of thin nets for low-resolution object recognition,” *TIP*, 2022.
- [38] T. Ma, W. Tian, and Y. Xie, “Multi-level knowledge distillation for low-resolution object detection and facial expression recognition,” *Knowledge-Based Systems*, 2022.
- [39] U. Michieli and P. Zanuttigh, “Knowledge distillation for incremental learning in semantic segmentation,” *CVIU*, 2021.
- [40] C. Saltori, F. Galasso, G. Fiameni, N. Sebe, F. Poiesi, and E. Ricci, “Compositional semantic mix for domain adaptation in point cloud segmentation,” *TPAMI*, 2023.
- [41] S. Guo, Y. Hu, J. M. Alvarez, and M. Salzmann, “Knowledge distillation for 6d pose estimation by aligning distributions of local predictions,” in *CVPR*, 2023.
- [42] Q. Guan, Z. Sheng, and S. Xue, “Hrpose: Real-time high-resolution 6d pose estimation network using knowledge distillation,” *Chinese Journal of Electronics*, 2023.
- [43] X. Lin, Y. Peng, L. Wang, X. Zhong, M. Zhu, J. Yang, C. Liu, and Q. Chen, “Cleanpose: Category-level object pose estimation via causal learning and knowledge distillation,” *arXiv:2502.01312*, 2025.
- [44] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “PointNet: Deep learning on point sets for 3d classification and segmentation,” in *CVPR*, 2017.
- [45] G. Qian, Y. Li, H. Peng, J. Mai, H. Hammoud, M. Elhoseiny, and B. Ghanem, “PointNeXT: Revisiting PointNet++ with Improved Training and Scaling Strategies,” in *NeurIPS*, 2022.
- [46] J. Masci, D. Boscaini, M. M. Bronstein, and P. Vandergheynst, “Geodesic Convolutional Neural Networks on Riemannian Manifolds,” in *ICCV-W*, 2015.
- [47] F. Monti, D. Boscaini, J. Masci, E. Rodol  , J. Svoboda, and M. M. Bronstein, “Geometric Deep Learning on Graphs and Manifolds using Mixture Model CNNs,” in *CVPR*, 2017.
- [48] M. Fey, J. E. Lenssen, F. Weichert, and H. M  ller, “SplineCNN: Fast Geometric Deep Learning with Continuous B-Spline Kernels,” in *CVPR*, 2018.
- [49] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, “KPConv: Flexible and deformable convolution for point clouds,” in *ICCV*, 2019.
- [50] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, “Dynamic graph CNN for learning on point clouds,” *ACM Transactions on Graphics*, 2019.
- [51] C. Choy, J. Gwak, and S. Savarese, “4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks,” in *CVPR*, 2019.
- [52] “Benchmark for 6d object pose estimation challenge 2024,” <https://bop.felk.cvut.cz/challenges/bop-challenge-2024/>, last accessed: February 24, 2025.
- [53] M. Oquab, T. Darcret, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al., “Dinov2: Learning robust visual features without supervision,” *arXiv:2304.07193*, 2023.
- [54] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *CACM*, 1981.
- [55] Y. Chen and G. Medioni, “Object modelling by registration of multiple range images,” *IVC*, 1992.
- [56] J. Gou, B. Yu, S. J. Maybank, and D. Tao, “Knowledge distillation: A survey,” *IJCV*, 2021.
- [57] Q.-Y. Zhou, J. Park, and V. Koltun, “Open3D: A modern library for 3D data processing,” *arXiv:1801.09847*, 2018.
- [58] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, “Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes,” in *ACCV*, 2012.
- [59] T. Hodan, P. Haluza, S. Obdrzalek, J. Matas, M. Lourakis, and X. Zabulis, “T-LESS: An RGB-D dataset for 6D pose estimation of texture-less objects,” in *WACV*, 2017.
- [60] T. Hodan, F. Michel, E. Brachmann, W. Kehl, A. G. Buch, D. Kraft, B. Drost, J. Vial, S. Ihrke, X. Zabulis, C. Sahin, F. Manhardt, F. Tombari, T.-K. Kim, J. Matas, and C. Rother, “BOP: Benchmark for 6D object pose estimation,” in *ECCV*, 2018.
- [61] A. Doumanoglou, R. Kouskouridas, S. Malassiotis, and T.-K. Kim, “Recovering 6D object pose and predicting next-best-view in the crowd,” in *CVPR*, 2016.
- [62] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, “PoseCNN: A Convolutional Neural Network for 6D object pose estimation in cluttered scenes,” in *Robotic Science and Systems*, 2018.
- [63] “Model-based 6d localization of unseen objects – bop-classic-core,” <https://bop.felk.cvut.cz/leaderboards/pose-estimation-unseen-bop23/bop-classic-core/>, last accessed: February 28, 2025.