Category-Level 6D Object Pose Estimation in Agricultural Settings Using a Lattice-Deformation Framework and Diffusion-Augmented Synthetic Data

Marios Glytsos[1], Panagiotis P. Filntisis[1], George Retsinas[1], Petros Maragos[1,2]

***Abstract*— Accurate 6D object pose estimation is essential for robotic grasping and manipulation, particularly in agriculture, where fruits and vegetables exhibit high intra-class variability in shape, size, and texture. The vast majority of existing methods rely on instance-specific CAD models or require depth sensors to resolve geometric ambiguities, making them impractical for real-world agricultural applications. In this work, we introduce PLANTPose, a novel framework for category-level 6D pose estimation that operates purely on RGB input. PLANTPose predicts both the 6D pose and deformation parameters relative to a base mesh, allowing a single category-level CAD model to adapt to unseen instances. This enables accurate pose estimation across varying shapes without relying on instance-specific data. To enhance realism and improve generalization, we also leverage Stable Diffusion to refine synthetic training images with realistic texturing, mimicking variations due to ripeness and environmental factors and bridging the domain gap between synthetic data and the real world. Our evaluations on a challenging benchmark that includes bananas of various shapes, sizes, and ripeness status demonstrate the effectiveness of our framework in handling large intraclass variations while maintaining accurate 6D pose predictions, significantly outperforming the state-of-the-art RGB-based approach MegaPose.**

I. INTRODUCTION

Robotics is transforming agriculture by offering scalable solutions for automated harvesting, reducing labor costs, and improving efficiency[1], [2]. At the core of these advances lies robotic grasping, which requires accurate 6D pose estimation of fruits and vegetables to enable precise picking and handling. However, unlike rigid industrial objects, fruits and vegetables exhibit significant natural variations in shape, size, and texture not only across different types but also within the same category due to growth, ripeness, and environmental conditions, making pose estimation particularly challenging.

Most existing 6D pose estimation methods rely on instance-level models, where a specific object must be known beforehand—either as a detailed CAD model [3], [4], [5], [6], [7] or through multiple reference images [4]. While effective in structured environments, these approaches are impractical for agriculture, where fruits and vegetables naturally vary in form, are encountered in novel configurations, and often appear amidst dense foliage or in close proximity to other produce. A more scalable alternative is category-level pose estimation, which generalizes across object instances within a class without requiring exact geometric models [8], [9]. However, current methods often depend on depth sensors to resolve geometric ambiguities, require complex non-differentiable solvers, or involve computationally expensive iterative refinements, limiting their deployment in real-world agricultural applications and purely RGB environments. As a result, achieving robust category-level pose estimation from RGB images alone remains challenging, particularly when large shape variability and real-time performance constraints are involved.

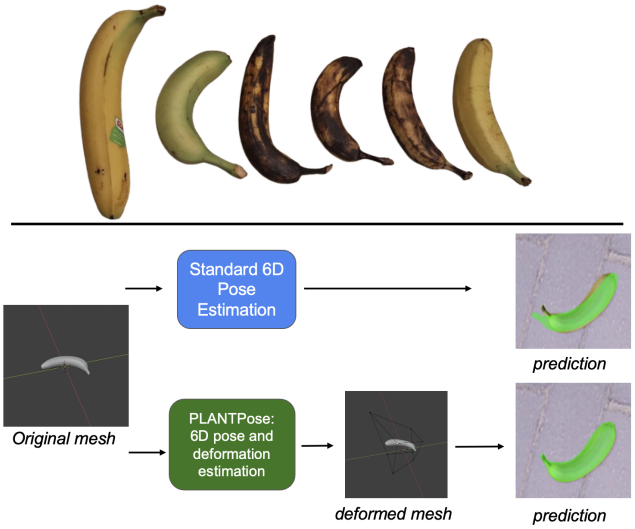

Fig. 1: Fruits exhibit significant intra-class variability in shape, size, and texture, posing challenges for standard 6D pose estimation methods. PLANTPose addresses this by simultaneously predicting both the deformation of a base mesh and the 6D pose, enabling more accurate and adaptable predictions.

Traditionally, 6D pose estimation methods rely on synthetic data for training, using simulated environments to generate large-scale datasets [10], [11], [12], [13]. While synthetic datasets provide controlled and scalable training, they often fail to capture the full spectrum of real-world variations, a limitation that is particularly pronounced in agricultural products. Fruits not only vary in shape and size but also undergo significant visual changes as they ripen, making robust 6D pose estimation even more challenging.

[1] M. Glytsos, P. P. Filntisis and G. Retsinas are with the Robotics Institute, Athena Research and Innovation Center, Maroussi 15125, Greece
[2] P. Maragos is with the Robotics Institute, Athena Research and Innovation Center, Maroussi 15125, Greece and the School of ECE, National Technical University of Athens, Greece

Moreover, collecting a comprehensive set of 3D models and textures for even a single object category is labor-intensive, impractical, and often infeasible.

To address these shortcomings, we introduce **PLANTPose** (**P**ose estimation using **L**attice deform**A**tio**N** for ca**T**egories), a novel framework for 6D pose estimation that employs intuitive deformations on base meshes to adapt a single category-level CAD model to unseen instances (see Fig. 1). By leveraging a compact set of deformation parameters, our method can capture broad intra-class shape variations *without the need for instance-specific models or depth data*. This makes PLANTPose particularly well-suited for categories such as agricultural produce, where high shape variability is the norm. **To further enhance realism and improve generalization to real-world imagery**, we employ Stable Diffusion-based image inpainting on our synthetic datasets, augmenting them with more realistic texturing to better mimic the diverse appearance of fruits at different stages of ripeness. This significantly narrows the domain gap between synthetic and real images, boosting pose estimation accuracy in real-world scenarios. We validate PLANTPose on the banana fruit, which naturally exhibits substantial shape and size deformations and undergoes significant visual changes as it transitions from unripe to ripe and eventually to rotten. This makes bananas an ideal test case for evaluating our framework. Extensive experiments demonstrate that PLANTPose achieves high pose estimation accuracy while generalizing well across diverse shapes, sizes, and textures, highlighting its effectiveness for real-world agricultural applications.

In short, our key contributions are:

- We present a novel framework for category-level 6D pose estimation of agricultural produce, leveraging intuitive deformations for flexible shape adaptation.
- We enhance the realism of synthetic datasets by leveraging Stable Diffusion with curated prompts and depth conditioning to capture the diverse textures of fruits at different ripeness stages.
- We compare our method against a widely used state-of-the-art 6D pose estimation approach, demonstrating that PLANTPose achieves significantly higher accuracy while effectively generalizing to diverse intra-class variations in fruit shapes and textures.

The source code and the synthetic dataset will be made publicly available.

II. Related Work

Category-level 6D pose estimation Category-level approaches predict the pose of previously unseen objects within a defined category. Many works adopt categorical mean shapes to facilitate feature alignment, improving robustness under intra-class variation. One of the early methods [8], introduced a canonical object space where point correspondences between input images and a normalized coordinate system are used to estimate pose. However, it struggled with large shape variations, leading to extensions such as SOCS [9], which introduces semantically-aware keypoint alignment, and NuNOCS [14], which supports non-uniform scaling for objects with varying aspect ratios. Several methods utilize shape priors to handle intra-class variations effectively. Tian et al. [15] proposed learning a categorical shape prior via an autoencoder and deforming it to match observed instances. Zhang et al. [16] improved upon this by introducing symmetry-aware shape prior deformation , allowing for direct pose regression while mitigating ambiguity in symmetric objects.

Implicit representations have also become popular in 6D pose estimation by providing continuous, differentiable shape modeling. ShAPO [17] jointly predicts shape, pose, and size using learned implicit fields, while [18] DISP6D separates shape and pose into distinct latent spaces for improved generalization. Neural Radiance Fields (NeRF) [19] have also been explored for category-level pose estimation, with approaches like NeRF-Pose [20] reconstructing object geometry before estimating pose. Unlike traditional correspondence-based methods, these techniques work in a continuous space, reducing alignment errors caused by discrete feature matching.

Recent advancements have focused on generalizing pose estimation beyond specific categories. FoundationPose [4] demonstrates strong performance on novel objects. Unlike previous category-specific methods, FoundationPose bridges model-based and model-free approaches by using implicit neural representations for novel view synthesis. While powerful, it still requires CAD models or multiple reference images, limiting its adaptability to truly unconstrained environments.

Synthetic Data for 6D Pose Estimation Many of the previous works in 6D pose estimation heavily rely on synthetic data due to the difficulty and time-consuming process of annotating real images or video datasets [4]. Most methods use CAD models or mesh models available in large scale 3D model databases [11], [12], [13], which store 3D object geometry with vertices, faces, and relative scale, making them suitable for rendering. To generate datasets, these models are placed in synthetic environments, such as Blender [21], where images can be rendered with ground truth annotations for position and rotation. This approach allows for large-scale dataset creation without manual annotation. Additionally, Anagnostopoulou et al. [22] leveraged Stable Diffusion with ControlNet to generate highly realistic synthetic data for mushrooms, demonstrating the effectiveness of diffusion-based approaches in modeling agricultural produce. In contrast to previous methods that rely solely on synthetic data, our generation pipeline combines traditional rendering with Stable Diffusion. We first generate physically plausible poses and scenes using physics-based simulation, and then enhance the results via img2img Stable Diffusion inpainting.

6D Pose Estimation for Agriculture Given the growing importance of 6D pose estimation in agricultural harvesting, numerous methods now address this challenge. Retsinas et al. [23] employed fully convolutional networks with implicit pose encodings to jointly perform mushroom segmentation and pose estimation, while Deep-ToMaTOS [24] introduced a deep learning framework that simultaneously predicts a tomato's ripeness level and its 6D pose. Li et al. [25] further

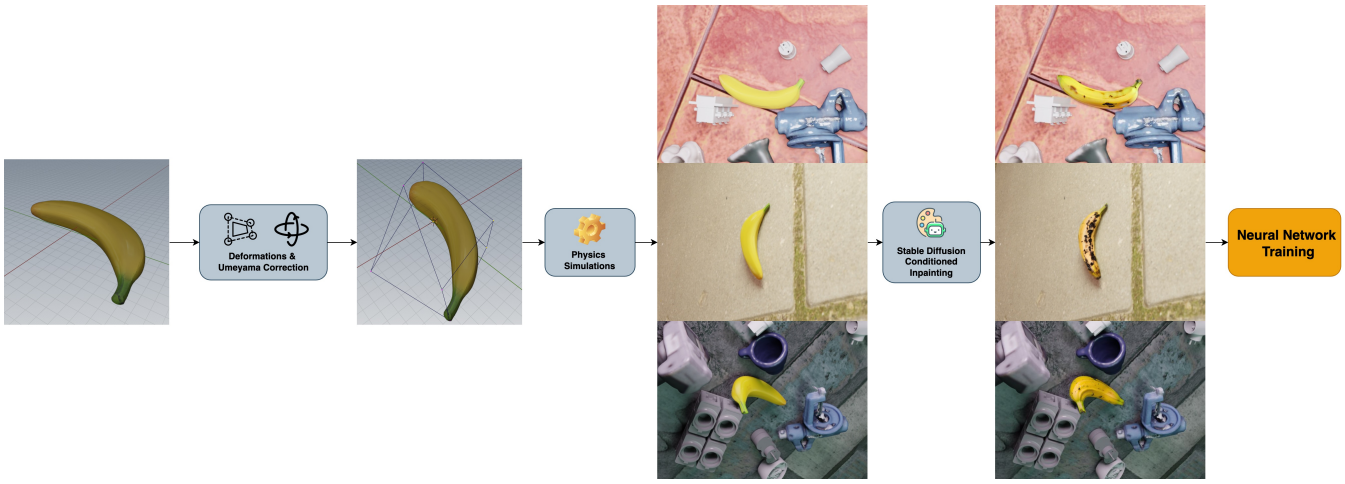

Fig. 2: Overview of the PLANTPose framework. Given a base mesh, we generate deformed object instances using lattice-based deformations. These instances are placed in physically plausible synthetic scenes rendered in Blender, which are further refined using Stable Diffusion inpainting to enhance texture realism. The generated dataset is then used to train a deep learning model for simultaneous 6D pose estimation and deformation prediction.

proposed a method for estimating both the 6D pose and 3D scale of strawberries, leveraging synthetic data and a YOLO-based architecture. Costanzo et al. [26] presented a system for apple grasping with an RGB-D camera, and an evaluation of different 6D pose estimation algorithms using RGB-D inputs for grasping was conducted in [27].

Although these approaches specifically target agricultural tasks, they primarily rely on adaptations of standard 6D pose pipelines and do not explicitly model object deformations. In contrast, our method is the first to estimate both the 6D pose and per-instance deformations—a crucial step toward accurately handling the natural shape variability of fruits and vegetables in real-world harvesting scenarios.

III. Method

The core concept behind PLANTPose is illustrated in Figure 2. Starting with a base mesh, we generate various deformations using lattice-based modeling (Sec. III-A). Next, we create physically plausible synthetic data with Blender, which is further refined using Stable Diffusion to enhance the texture realism of the object of interest (Sec. III-B). Finally, we train a deep learning model on the resulting dataset (Sec. III-C). The following sections describe each step in detail.

A. Lattice Deformations

In order to model intuitively the deformations around the object, we use lattices. In computational geometry, lattices serve as a framework for space partitioning, enabling efficient representation and transformation of spatial data [28].

The concept of lattice-based deformations was introduced in computer graphics to provide a structured approach for manipulating 3D objects smoothly. A lattice consists of a set of control points arranged in a grid, where each point influences the space around it. The term "lattice" in this context refers to its structured, grid-like nature, which allows for spatial transformations. The deformation of any given point inside the lattice is controlled by interpolation between control points. In our case, the bounding box of the object is the lattice that controls the deformation. This way, we control the deformation by the 3D movement of the bounding box boundaries (i.e., the control points). In other words, we can simulate useful deformations using only 8×3 parameters. Despite its simplicity, such an approach is able to create non-trivial variations while assisting the formulation of the estimation step, since the developed neural network **can detect deformation as a regressor of a fixed-sized feature**. To avoid extreme, non-useful cases, we constraint the magnitude of the deformations to an empirical fixed upper bound.

Having defined the control points, one should perform the deformation in the lattice interior using an interpolation step, typically *linear interpolation* or *B-spline interpolation*. We focus on *cubic B-spline* interpolation, which provides smoother (C^2) deformations.

Let $\mathbf{box\_min} = (\min_x, \min_y, \min_z)$ and $\mathbf{box\_max} = (\max_x, \max_y, \max_z)$ be the bounding box corners. A mesh vertex $\mathbf{p} = (p_x, p_y, p_z)$ is mapped to $\mathbf{u} = (u, v, w) \in [0, 1]^3$ so each (u, v, w) is the fractional distance along the x, y, z axes. A cubic B-spline lattice is determined by eight corner offsets $\mathbf{C}_{i,j,k} \in \mathbb{R}^3$ for $(i, j, k) \in \{0, 1\}^3$. These corner offsets can be extended (clamped) to a $4 \times 4 \times 4$ grid for correct boundary behavior.

The 1D B-spline basis functions for $t \in [0, 1]$ are

$$W_0(t) = \frac{1 - 3t + 3t^2 - t^3}{6}, \quad W_1(t) = \frac{4 - 6t^2 + 3t^3}{6} \tag{1}$$

$$W_2(t) = \frac{(1 + 3t + 3t^2 - 3t^3)}{6}, \quad W_3(t) = \frac{t^3}{6}. \tag{2}$$

They are evaluated at u, v, w along the x, y, z axes, respectively, giving $W_i(u), W_j(v), W_k(w)$. A vertex's displacement is computed by summing over the three coordinate axes

separately. Let each control point $\mathbf{C}_{i,j,k}$ have components $\big(C^x_{i,j,k},\, C^y_{i,j,k},\, C^z_{i,j,k}\big)$. Then, for $(u, v, w) \in [0, 1]^3$,

$$\delta_x(u,v,w) = \sum_{i=0}^{3}\sum_{j=0}^{3}\sum_{k=0}^{3} W_i(u)\, W_j(v)\, W_k(w)\; C^x_{i,j,k},$$

$$\delta_y(u,v,w) = \sum_{i=0}^{3}\sum_{j=0}^{3}\sum_{k=0}^{3} W_i(u)\, W_j(v)\, W_k(w)\; C^y_{i,j,k},$$

$$\delta_z(u,v,w) = \sum_{i=0}^{3}\sum_{j=0}^{3}\sum_{k=0}^{3} W_i(u)\, W_j(v)\, W_k(w)\; C^z_{i,j,k}.$$

These form the displacement vector $\boldsymbol{\delta}(u, v, w) = \big[\delta_x, \delta_y, \delta_z\big]$, and the deformed position becomes $\mathbf{p}' = \mathbf{p} + \boldsymbol{\delta}(u, v, w)$.

Because B-splines ensure C^2 continuity in each dimension, they produce more natural, smoothly varying deformations, making them preferable for applications requiring realistic shape modifications.

B. Synthetic Data Generation

Creating a real large-scale dataset that adequately captures intra-class variations with deformation data and includes accurate 3D annotations of the pose/deformation is infeasible in practice. Therefore, we rely on synthetic data generation to cover a wide range of shape and pose variations. For this, we use BlenderProc [21], a Python-based API for Blender, to simulate physically plausible object placement and rendering.

Our synthetic dataset is generated by constructing virtual environments with 3D object models, where target objects are placed alongside distractors to introduce occlusions. Scenes include procedural room geometry (walls, floors) for context, textured surfaces for realism, and lighting variations (ambient and directional) to simulate diverse illumination conditions. To ensure physically plausible positioning, we utilize a physics simulation where objects are dropped into the scene and settle naturally based on their physical properties. Once the scene is stable, we define a point of interest and sample multiple camera poses around it. Each camera view follows randomized extrinsic parameters (position and orientation) while maintaining visibility of the target object. The intrinsic parameters are fixed to simulate a real-world camera. Images are then rendered from multiple angles, capturing variations in occlusion, lighting, and perspective.

Introducing Lattice-Based Deformations We now modify the previously described synthetic data pipeline and introduce lattice-based deformations with the following procedure: We first place a tight lattice bounding box around the object of interest. Then we randomly perturb the control points of the lattice within an empirically set constrained range. Our goal is to have a unique set of annotations $\{\mathbf{t}, \mathbf{R}, \delta\}$, where $\mathbf{t}$ is the translation, $\mathbf{R}$ the rotation, and δ the deformation. However, the B-spline interpolation, contrary to the linear one, may introduce unwanted global translation and rotation, depending on the random pertubation. To alleviate this amgibuity, after deforming the object, we apply the Umeyama algorithm [29], and compute an optimal similarity transformation between the original mesh and the deformed one:

$$(s, \mathbf{R}, \mathbf{t}) = \text{Umeyama}(\mathbf{P}_{\text{original}}, \mathbf{P}_{\text{deformed}}), \tag{3}$$

Here, the function Umeyama estimates the optimal similarity transformation (scaling factor s - which we fix to 1, rotation matrix $\mathbf{R}$, and translation vector $\mathbf{t}$) that best aligns the set of original points $\mathbf{P}_{\text{original}}$ with the deformed points $\mathbf{P}_{\text{deformed}}$ in a least-squares sense. The lattice points are then corrected to ensure that deformations remain independent of pose transformations.

Enhancing realism with Stable Diffusion

As mentioned in the introduction, synthetic data pipelines, like the one described so far, do not fully capture the texture variations and realism found in real-world settings. This limitation is particularly evident in agricultural products, where surface textures vary significantly due to ripeness, environmental conditions, and natural inconsistencies. Previous approaches, such as FoundationPose[4], addressed this issue by introducing texture variations directly on 3D models before rendering. However, this approach still involves the rendering process, leaving a small but non-negligible domain gap between synthetic and real-world data.

In contrast, we enhance realism after rendering by applying Stable Diffusion inpainting directly to the final image, modifying only the object of interest while preserving the background and scene consistency. To ensure that the object's pose remains unchanged, we condition the Stable Diffusion generation using ControlNet [30], leveraging depth maps from the rendering pipeline. Finally, we curate a set of prompts describing various textures and color variations across different stages of a fruit's lifecycle—including unripe, ripe, and rotten appearances, as well as natural color variations. This approach minimizes the domain gap while maintaining geometric and pose consistency, leading to improved generalization in real-world agricultural applications.

In summary, we aim to acquire a set of very realistic images via the synthetic scenes, while retaining the 3D annotations required for training a network.

C. Network and Training

Given the synthetic dataset with the 3D annotations, we can now train a neural network for detecting both the 6D pose and the deformation. The input of the network is the cropped **RGB** (*we do not use depth*) image of the desired object. During training we use the projected 2D vertices of the object to create the bounding box. For inference, we train a YOLOv11[31] model on our dataset.

For our neural network model we use a small ViT[32] backbone with $30.9M$ parameters as a feature encoder which uses 32x32 patches, pretrained on ImageNet-21k. After the ViT encoder, we employ three lightweight heads:

Rotation Head: predicts the 6D rotation representation [33] that is then orthonormalized into a valid 3×3 rotation matrix.

Translation Head: outputs a 3D vector $\mathbf{t}$ representing the object's translation in the cropped image frame.

Deformation Head: produces a 24D offset vector that warps a template mesh via a $2 \times 2 \times 2$ lattice.

Training Losses. Let $\mathbf{r}_{6d}$ be the predicted rotation (in 6D), $\mathbf{t}$ the translation, and δ the lattice offsets. We supervise them with:

- **Rotation Loss:** Mean squared error (MSE) on $\mathbf{r}_{6d}$ to match the ground truth rotation.
- **Deformation Loss:** MSE on δ, comparing them to the synthetic deformations used in data generation.
- **2D Projection Loss:** to auxiliary supervise the estimated set of parameters $\{\mathbf{t}, \mathbf{R}, \delta\}$ and essentially derive translation parameters, we project the deformed and translated/rotated 3D vertices (i.e., the estimated final mesh) into the cropped image and measure their mean squared error against 2D keypoints of the ground-truth mesh.

Formally,

$$\begin{aligned}\mathcal{L} \; &= \lambda_r \, \|\mathbf{r}_{6d}^{\text{pred}} - \mathbf{r}_{6d}^{\text{gt}}\|^2 \; + \; \lambda_d \, \|\delta^{\text{pred}} - \delta^{\text{gt}}\|^2 \\ &+ \; \lambda_p \, \|\text{Proj}(\mathbf{P}|\mathbf{R}, \mathbf{t}, \delta) - \mathbf{v}_{2d}^{\text{gt}}\|^2.\end{aligned} \tag{4}$$

This combination aligns the rotation, warping, and final 2D alignment, jointly driving pose and shape accuracy. We train the network end-to-end with Adam optimizer [34] on cropped object patches, applying standard image augmentations to improve robustness. We train on 4,000 synthetic scenes, each with a random deformation, capturing three images per scene from different camera positions.

D. From cropped to full image

Since the input to the network is the cropped object of interest, during inference we store the offset $(x_{\min}, y_{\min})$ and scale factor used for cropping and resizing. After predicting the pose in the cropped frame, we *undo* these transformations to re-project the object's 2D vertices back into full-image coordinates. From there, we apply a PnP solver [35] using our lattice-deformed 3D points and the intrinsic parameters of the camera to find the translation $\mathbf{t}$ with respect to the camera frame.

Fig. 3: Qualitative results on the banana test benchmark comparing MegaPose with PlantPose. The final column shows the predictions in 3D space (with a different angle to facilitate comparison): the green color denotes the ground truth mesh and 6D pose, blue is the result of PlantPose, and Red is the result of MegaPose.

IV. Results and Evaluation

Benchmark Dataset and Metrics To validate our framework, we collected an in-house dataset of six bananas with distinct shapes and varying ripeness stages (see top part of Fig. 1). We captured 100 images using an Intel RealSense D435 and manually annotated them, after first 3D scanning each banana using an iPhone 14 Pro with a LiDAR sensor. We evaluate our method using the following metrics: (a) Chamfer Distance, which measures the geometric discrepancy between the ground truth and predicted meshes, as the scanned meshes lack per-vertex correspondence with our base banana model; (b) Mean and Median Rotation Error; (c) Mean and Median Translation Error; and (d) Deformation Error. To compute rotation and translation errors, we align the ground truth scanned meshes with the base banana mesh using ICP. The deformation error is then computed after removing the estimated rotation and translation, isolating the deformation component.

Comparison with State-of-the-Art. We compare our approach with MegaPose[3], which, like our method, relies solely on RGB input. In contrast, other category-level pose estimation methods ([9], [8], [14], [36], [37]), to the best of our knowledge, rely on depth information (sometimes additionally to RGB). This highlights a key distinction, as our method achieves competitive performance without requiring depth, demonstrating its effectiveness as a purely vision-based solution. We use the publicly available implementation of MegaPose and use as input the same banana mesh for all images. We present results on our benchmark dataset in Table I. As we can see, PLANTPose significantly outperforms MegaPose across all metrics.

TABLE I: Comparison of PLANTPose with MegaPose on the banana benchmark dataset. For all metrics, lower is better.

Method	Chamfer (mm) ↓	Rot. (deg) ↓		Trans. (mm) ↓		Deform. (mm) ↓
	Dist.	Mean	Med.	Mean	Med.	Error
MegaPose [3]	90.1	52.4	43.6	59.9	45.6	29.1
PLANTPose (Ours)	**59.8**	**32.6**	**23.6**	**42.5**	**37.9**	**12.1**

Figure 3 presents qualitative results from our test set. As shown, MegaPose struggles with accurately predicting the rotation and translation of the banana, as it relies on an average banana mesh to explain the visual scene and lacks the ability to model deformations. This limitation is particularly evident in the final column, where the results are visualized in 3D space. In contrast, PLANTPose successfully captures both the object's deformation and its 6D pose, leading to more precise translation and rotation predictions in the 3D world.

Ablation studies We conduct ablation studies to evaluate the effectiveness of our Stable Diffusion-based realism enhancement and the impact of omitting the Umeyama step for correcting deformation-induced rotation and translation. The results of these experiments are presented in Table II. As evident from the results, each component plays a crucial role in achieving optimal performance. The Umeyama correction effectively resolves ambiguities in rotation and translation introduced by the initial deformation, while the Stable Diffusion-enhanced dataset significantly improves generalization.

TABLE II: Ablation study results evaluating the impact of different components on pose estimation accuracy. Chamfer distance (Dist.), rotation (Rot.), and translation (Trans.) errors are reported as mean (Mean) and median (Med.), along with the deformation error.

Method	Chamfer (mm) ↓	Rot. (deg) ↓		Trans. (mm) ↓		Deform. (mm) ↓
	Dist.	Mean	Med.	Mean	Med.	Error
PLANTPose (Full)	**59.8**	**32.6**	23.6	**42.5**	**37.9**	**12.1**
w/o SD	71.5	37.7	**23.0**	48.9	48.7	13.9
w/o Umeyama	73.1	36.5	26.2	50.3	45.8	12.7
w/o Umeyama and SD	89.7	36.1	26.7	61.1	61.9	26.5

Figure 4 shows that without Umeyama, the model struggles with translation because deformation, rotation, and translation are not disentangled—deformations inherently introduce unwanted pose changes. Without Stable Diffusion, the model performs well on familiar banana textures but fails on unseen ones, losing the overall pose.

Limitations While our method accurately predicts both the deformation parameters and 6D pose of an object, it has several limitations. First, it currently requires training a separate model for each object category. A more general solution would accept a base CAD model as an additional input, adapting across multiple categories without retraining and still preserving mesh deformation capabilities. Additionally, in rare cases, even with depth conditioning, Stable Diffusion may slightly alter the object's 6D pose. Exploring more robust approaches for conditioning generative models could further enhance our synthetic data pipeline.

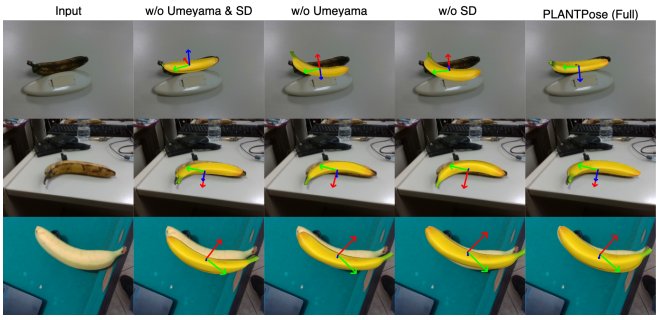

Fig. 4: Qualitative ablation results from different model variations on the banana test benchmark. Each row shows a different input, while columns compare results from different models.

V. Conclusion

We introduced PLANTPose, a category-level 6D pose estimation framework that predicts both pose and deformation parameters from RGB images. By leveraging lattice-based deformations and Stable Diffusion-based texture augmentation, our method enables accurate pose estimation across diverse object instances without requiring instance-specific models or depth input. Our experiments demonstrate strong performance on a banana benchmark, significantly outperforming the state-of-the-art method MegaPose. For future work, we aim to expand our approach to multiple object categories beyond fruits and incorporate the base CAD model as an input, allowing greater adaptability across different object types.

References

[1] G. Kootstra, X. Wang, P. M. Blok, J. Hemming, and E. Van Henten, "Selective harvesting robotics: current research, trends, and future directions," *Current Robotics Reports*, vol. 2, pp. 95–104, 2021.

[2] K. Zhang, K. Lammers, P. Chu, N. Dickinson, Z. Li, and R. Lu, "Algorithm design and integration for a robotic apple harvesting system," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 9217–9224.

[3] Y. Labbé, 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," *arXiv preprint arXiv:2212.06870*, 2022.

[4] W. Bowen, Y. Wei, K. Jan, and B. Stan, "FoundationPose: Unified 6d pose estimation and tracking of novel objects," in *CVPR*, 2024.

[5] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," in *Proceedings of Robotics: Science and Systems*, 2018.

[6] Y. Labbe, J. Carpentier, M. Aubry, and J. Sivic, "Cosypose: Consistent multi-view multi-object 6d pose estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[7] V. N. Nguyen, T. Groueix, M. Salzmann, and V. Lepetit, "Gigapose: Fast and robust novel object pose estimation via one correspondence," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 9903–9913.

[8] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L. J. Guibas, "Normalized object coordinate space for category-level 6d object pose and size estimation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[9] B. Wan, Y. Shi, and K. Xu, " SOCS: Semantically-aware Object Coordinate Space for Category-Level 6D Object Pose Estimation under Large Shape Variations ," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 14 019–14 028.

[10] T. Hodaň, F. Michel, E. Brachmann, W. Kehl, A. Glent Buch, D. Kraft, B. Drost, J. Vidal, 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," *European Conference on Computer Vision (ECCV)*, 2018.

[11] A. X. Chang, T. A. Funkhouser, L. J. Guibas, P. Hanrahan, Q.-X. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "Shapenet: An information-rich 3d model repository," *CoRR*, vol. abs/1512.03012, 2015.

[12] 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 *International Conference on Robotics and Automation (ICRA)*, 2022, pp. 2553–2560.

[13] M. Deitke, D. Schwenk, J. Salvador, L. Weihs, O. Michel, E. VanderBilt, L. Schmidt, K. Ehsani, A. Kembhavi, and A. Farhadi, "Objaverse: A universe of annotated 3d objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 13 142–13 153.

[14] B. Wen, W. Lian, K. Bekris, and S. Schaal, "Catgrasp: Learning category-level task-relevant grasping in clutter from simulation," *ICRA 2022*, 2022.

[15] M. Tian, M. H. Ang Jr, and G. H. Lee, "Shape prior deformation for categorical 6d object pose and size estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, August 2020.

[16] R. Zhang, Y. Di, F. Manhardt, N. Navab, F. Tombari, and X. Ji, "Ssp-pose: Symmetry-aware shape prior deformation for direct category-level object pose estimation," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (Accepted) (IROS)*, 2022.

[17] M. Z. Irshad, S. Zakharov, R. Ambrus, T. Kollar, Z. Kira, and A. Gaidon, "Shapo: Implicit representations for multi-object shape appearance and pose optimization," in *European Conference on Computer Vision (ECCV)*, 2022.

[18] Y. Wen, X. Li, H. Pan, L. Yang, Z. Wang, T. Komura, and W. Wang, "Disp6d: Disentangled implicit shape and pose learning for scalable 6d pose estimation," in *European Conference on Computer Vision (ECCV)*, 2022.

[19] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *ECCV*, 2020.

[20] F. Li, H. Yu, I. Shugurov, B. Busam, S. Yang, and S. Ilic, "Nerf-pose: A first-reconstruct-then-regress approach for weakly-supervised 6d object pose estimation," 2023. [Online]. Available: https://arxiv.org/abs/2203.04802

[21] M. Denninger, D. Winkelbauer, M. Sundermeyer, W. Boerdijk, M. Knauer, K. H. Strobl, M. Humt, and R. Triebel, "Blenderproc2: A procedural pipeline for photorealistic rendering," *Journal of Open Source Software*, vol. 8, no. 82, p. 4901, 2023. [Online]. Available: https://doi.org/10.21105/joss.04901

[22] D. Anagnostopoulou, G. Retsinas, N. Efthymiou, P. Filntisis, and P. Maragos, "A realistic synthetic mushroom scenes dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 6282–6289.

[23] G. Retsinas, N. Efthymiou, and P. Maragos, "Mushroom segmentation and 3d pose estimation from point clouds using fully convolutional geometric features and implicit pose encoding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 6264–6271.

[24] J. Kim, H. Pyo, I. Jang, J. Kang, B. Ju, and K. Ko, "Tomato harvesting robotic system based on deep-tomatos: Deep learning network using transformation loss for 6d pose estimation of maturity classified tomatoes with side-stem," *Computers and Electronics in Agriculture*, vol. 201, p. 107300, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168169922006123

[25] L. Li and H. Kasaei, "Single-shot 6dof pose and 3d size estimation for robotic strawberry harvesting," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 4988–4993.

[26] M. Costanzo, M. De Simone, S. Federico, C. Natale, and S. Pirozzi, "Enhanced 6d pose estimation for robotic fruit picking," in *2023 9th International Conference on Control, Decision and Information Technologies (CoDIT)*. IEEE, 2023, pp. 901–906.

[27] A. B. Alterani, M. Costanzo, M. De Simone, S. Federico, and C. Natale, "Experimental comparison of two 6d pose estimation algorithms in robotic fruit-picking tasks," *Robotics*, vol. 13, no. 9, 2024. [Online]. Available: https://www.mdpi.com/2218-6581/13/9/127

[28] C. D. Toth, J. O'Rourke, and J. E. Goodman, *Handbook of discrete and computational geometry*. CRC press, 2017.

[29] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 4, pp. 376–380, 1991.

[30] L. Zhang, A. Rao, and M. Agrawala, "Adding conditional control to text-to-image diffusion models," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 3836–3847.

[31] G. Jocher and J. Qiu, "Ultralytics yolo11," 2024. [Online]. Available: https://github.com/ultralytics/ultralytics

[32] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *ICLR*, 2021.

[33] M. Zhu, K. G. Derpanis, Y. Yang, S. Brahmbhatt, M. M. Zhang, C. J. Phillips, M. Lecce, and K. Daniilidis, "Single image 3d object detection and pose estimation for grasping," *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3936–3943, 2014.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[35] V. Lepetit, F. Moreno-Noguer, and P. Fua, "epnp: An accurate o(n) solution to the pnp problem," *International Journal of Computer Vision*, vol. 81, no. 2, p. 155, 2009.

[36] Y. Di, R. Zhang, Z. Lou, F. Manhardt, X. Ji, N. Navab, and F. Tombari, "Gpv-pose: Category-level object pose estimation via geometry-guided point-wise voting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 6781–6791.

[37] J. Lin, Z. Wei, Z. Li, S. Xu, K. Jia, and Y. Li, "Dualposenet: Category-level 6d object pose and size estimation using dual pose network with refined learning of pose consistency," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3560–3569.