

IMIGRASP: Zero-shot 6-DOF pose estimation for grasping via GenAI imagination and cross-image geometric transfer

Muhammad Aun Raza^{*1}

¹Department of Electrical Engineering, Pakistan Institute of Engineering and Applied Sciences (PIEAS), Islamabad, Pakistan

Abstract Robotic grasping in unstructured environments remains a fundamental challenge, particularly for novel objects not encountered during training. Traditional approaches rely on large scale annotated datasets, extensive training, and hand crafted features. This work introduces IMIGRASP (Imagine to Grasp), a zero-shot framework for 6-DOF pose estimation that leverages generative artificial intelligence for imagination guided pose estimation. The key insight is that modern generative models, trained on vast internet scale visual data, implicitly encode rich knowledge about human grasping behavior and object affordances. Given a single RGB image of a target object, the pipeline employs GPT-5.6 Luna to generate a corresponding image with a realistic hand grasping the object, complete with colored landmarks at the thumb tip, index fingertip, and wrist. SAM3 is then used for precise segmentation of objects and hands. The main technical contribution of this work lies in the geometric layer that converts this imagined interaction into an executable grasp: a color-LUT landmark detector with mask-constrained refinement, a mask-validation and expanding-circle correction step, a 2D geometric construction that determines the grasp point from the imagined hand configuration, a scale-invariant bounding-box transfer that maps landmarks from the generated image to the real image, an RGB-D based 3D reconstruction of the grasp frame, and a rule-based alignment scheme that selects the gripper orientation from the reconstructed geometry. Together these components form a fully deterministic, training-free geometric transfer pipeline. The approach is validated on 130 diverse household objects, achieving a success rate of 86.9% without any prior training or object specific tuning. The results demonstrate that generative AI models can serve as powerful zero-shot priors for robotic manipulation when coupled with principled geometric transfer.

Keywords: Generative AI for pose estimation, zero-shot robotic grasping, 6-DOF grasp generation, cross-image geometric transfer, foundation models for manipulation

1. Introduction

Robotic grasping is a cornerstone capability for autonomous manipulation systems, enabling robots to interact with and manipulate objects in unstructured environments [1, 2]. Despite significant advances in recent years, robust and generalizable grasping remains challenging, particularly for novel objects and poses not seen during training [3]. Recent approaches have explored zero-shot grasping using generative examples and semantic guidance [4]. In this work, IMIGRASP (Imagine to Grasp) is introduced, a zero-shot framework that leverages generative AI for 6-DOF grasp generation. The core insight is that GPT-5.6 Luna’s ability to produce realistic hand object interactions from a single RGB image can be harnessed to infer optimal grasp configurations without any prior training [5, 6], since modern generative models trained

^{*}Corresponding author. E-mail: muhammadaun33555@gmail.com

on vast internet scale data implicitly encode rich knowledge about human behavior, object affordances, and physical interactions [7, 8, 9]. The main technical contribution of IMIGRASP is the geometric layer that converts this imagined hand-object interaction into an executable grasp: a color-LUT landmark detector constrained to the segmented hand mask extracts thumb tip, index fingertip, and wrist from the generated image, followed by a mask-validation and expanding-circle correction step that guarantees the landmarks lie on the object; a purely 2D geometric construction—a line through the wrist parallel to the fingertip segment, its intersection with the perpendicular bisector of the fingertips, and a boundary walk toward the object mask—yields the grasp point without any learning; a scale-invariant bounding-box transfer maps all landmarks from the generated image to the real image using the diagonal ratio of the corresponding object masks, making the correspondence robust to changes in scale and position; and a rule-based alignment scheme reconstructs the full grasp frame from the relative geometry of the fingertips, wrist, and grasp point after RGB-D back-projection. The complete pipeline thus consists of generative image synthesis with colored landmarks, segmentation using SAM3, color based landmark detection, 2D geometric construction for grasp point identification, scale-invariant bounding box transfer, RGB-D based 3D reconstruction, and rule-based grasp frame computation and execution [10, 11, 12]. Recent zero-shot and task-oriented grasping approaches have also been investigated [13, 14, 15]. Traditional approaches rely on large scale annotated datasets and extensive training, which limits their applicability to new domains and objects [16, 17, 18]. Affordance learning and language grounding have been explored for robotic manipulation [19, 20, 21]. The concept of imagination has also been investigated in robotics for planning and reasoning [22, 23, 24]. In contrast to these approaches, the proposed framework is unique in using generative AI capabilities for zero-shot grasp generation, requiring no training or fine-tuning, and in coupling this generative prior with a closed-form geometric transfer layer [25, 26, 27]. The main contributions include: a zero-shot framework for 6-DOF grasp generation using generative AI without any training data; a geometric transfer pipeline comprising color-LUT landmark detection, mask-validation and correction, 2D geometric construction of the grasp point, scale-invariant bounding-box transfer, and rule-based grasp frame alignment; a pipeline combining GPT-5.6 Luna, SAM3, and this geometric layer for grasp pose estimation; and comprehensive experimental validation on 130 diverse household objects with a Franka Panda robot [28, 29, 30]. A supplementary video demonstrating the complete pipeline is available as multimedia material.

2. Related Work

2.1. GenAI for Grasping

Recent works have explored generative AI for robotic grasping. GRIM [1] retrieves and re-aligns grasps from generative examples in a training-free memory-based pipeline. SegGrasp [2] employs semantic and geometric guided segmentation for zero-shot task-oriented grasping. Imagine2Act [3] leverages object-action motion consistency from imagined goals within a trained imitation-learning policy. Affordance diffusion [4] synthesizes hand-object interactions using diffusion models trained on large-scale data. GraspGen-X [11] performs cross-embodiment diffusion-based grasping trained on billions of simulated grasps. GraspLDP [12] proposes a latent diffusion approach for generalizable grasping. GraspFoM [13] utilizes 3D foundation priors for joint reconstruction and grasp generation. DCGrasp [14] enables controllable, diffusion-based hand-object grasp generation. DextER [15] employs autoregressive, language-driven dexterous grasp generation with embodied contact reasoning. Cross-view fusion [16] fuses multiple camera views via a trained network for robust 6-DoF pose estimation, while Contact-GraspNet [10] serves as a purely discriminative, non-generative baseline trained on synthetic point clouds.

2.2. Imagination and Affordance Learning

The concept of imagination and affordance learning has been explored for robotic manipulation. Luo et al. [26] learn affordance grounding from exocentric images via a trained cross-view transfer framework. Part-SLIP [27] lifts 2D detections from a pretrained image-language model for low-shot 3D part segmentation. SATR [28] performs zero-shot 3D shape segmentation using off-the-shelf 2D detectors, requiring no training. CLIP-based models [29] enable open-vocabulary understanding through large-scale contrastive pretraining. Visual foresight [30] learns physical dynamics through video prediction, requiring extensive training data. One-shot affordance learning [18] trains on a single example per category yet generalizes reasonably to unseen objects and affordances. ShapeGrasp [19] uses large language models over a geometric decomposition for zero-shot, training-free grasp reasoning. Language-guided dexterous grasp generation [20] guides grasp selection via a dataset-trained model, while AnyGrasp [17] offers a purely discriminative grasp-perception baseline with no affordance or imagination component.

3. Methodology

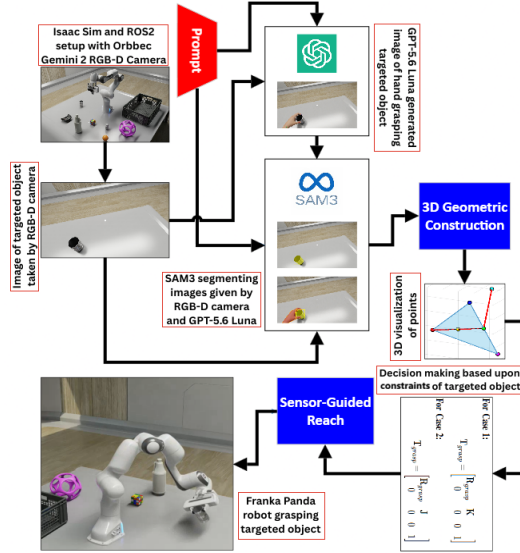


Figure 1: Block diagram illustrating the complete IMIGRASP pipeline for zero-shot 6-DOF grasp generation.

3.1. GPT-5.6 Luna Image Generation

Given the original RGB image $\mathbf{I}_{orig} \in \mathbb{R}^{1080 \times 1920 \times 3}$, GPT-5.6 Luna generates a corresponding image $\mathbf{I}_{gpt}$ with a realistic hand grasping the object, preserving object position, orientation, and scale. The prompt requires red, blue, and green circles at thumb tip (A), index fingertip (B), and wrist (C), respectively. This step leverages the generative model’s understanding of human grasping behavior to produce physically plausible hand configurations without any training data.

Formally, let $\mathcal{G}$ denote the generative mapping performed by GPT-5.6 Luna. The generated image is expressed as

$$\mathbf{I}_{gpt} = \mathcal{G}(\mathbf{I}_{orig}, \mathcal{P}), \quad (1)$$

where $\mathcal{P}$ is the structured prompt encoding the grasping task and the landmark color specification. The prompt $\mathcal{P}$ is composed of three components:

$$\mathcal{P} = \{\mathcal{P}_{task}, \mathcal{P}_{landmark}, \mathcal{P}_{constraint}\}, \quad (2)$$

where $\mathcal{P}_{task}$ describes the grasping action, $\mathcal{P}_{landmark}$ specifies the red, blue, and green circles at the thumb tip, index fingertip, and wrist, respectively, and $\mathcal{P}_{constraint}$ enforces preservation of the object’s pose and scale. The preservation constraint can be expressed as

$$\mathcal{L}_{preserve} = \|\mathbf{I}_{orig} \odot \mathcal{M}_{obj} - \mathbf{I}_{gpt} \odot \mathcal{M}_{obj}\|_2^2, \quad (3)$$

where $\mathcal{M}_{obj}$ denotes the object mask in the original image and $\odot$ denotes element-wise multiplication. This constraint ensures that the object region remains unaltered while the hand is synthesized in its vicinity. The landmark specification $\mathcal{P}_{landmark}$ instructs the model to render three distinct colored markers at specific anatomical locations:

$$\mathcal{P}_{landmark} = \{(\text{red}, \mathbf{p}_{thumb}), (\text{blue}, \mathbf{p}_{index}), (\text{green}, \mathbf{p}_{wrist})\}, \quad (4)$$

where $\mathbf{p}_{thumb}$, $\mathbf{p}_{index}$, and $\mathbf{p}_{wrist}$ denote the target pixel locations of the thumb tip, index fingertip, and wrist, respectively. The generated image $\mathbf{I}_{gpt}$ thus contains a hand configuration that is both physically plausible and semantically consistent with the object’s geometry, without requiring any training data or explicit hand pose annotations.

3.2. Segmentation and Landmark Detection

SAM3 [9] segments object masks $\mathcal{M}_{gpt}$ and $\mathcal{M}_{real}$, and hand mask $\mathcal{M}_{hand}$. Colored landmarks are detected using a color LUT approach. For each target color, a LUT is built and the point is found by minimizing within the hand mask:

$$\text{LUT}_{color}(r, g, b) = (r - r_{target})^2 + (g - g_{target})^2 + (b - b_{target})^2 \quad (5)$$

$$P = \arg \min_{\mathbf{p} \in \mathcal{M}_{hand}} \text{LUT}_{color}(\mathbf{I}_{gpt}(\mathbf{p})) \quad (6)$$

This yields $A = (u_A, v_A)$ for thumb tip, $B = (u_B, v_B)$ for index fingertip, and $C = (u_C, v_C)$ for wrist.

More precisely, let $\mathcal{C} = \{\text{red}, \text{blue}, \text{green}\}$ denote the set of target colors, with corresponding RGB triplets $\mathbf{c}_{red} = (r_{red}, g_{red}, b_{red})$, $\mathbf{c}_{blue} = (r_{blue}, g_{blue}, b_{blue})$, and $\mathbf{c}_{green} = (r_{green}, g_{green}, b_{green})$. For each color $c \in \mathcal{C}$, the LUT is defined as

$$\text{LUT}_c(r, g, b) = (r - r_c)^2 + (g - g_c)^2 + (b - b_c)^2, \quad (7)$$

which measures the squared Euclidean distance in RGB space between a given pixel color and the target color $\mathbf{c}_c$. The landmark detection for color c is then formulated as

$$\mathbf{p}_c^* = \arg \min_{\mathbf{p} \in \mathcal{M}_{hand}} \text{LUT}_c(\mathbf{I}_{gpt}(\mathbf{p})), \quad (8)$$

where $\mathbf{I}_{gpt}(\mathbf{p})$ denotes the RGB triplet at pixel $\mathbf{p} = (u, v)$ in the generated image. The hand mask $\mathcal{M}_{hand}$ restricts the search to the hand region, preventing false detections from background pixels. The three landmarks are thus obtained as

$$A = \mathbf{p}_{red}^* = (u_A, v_A), \quad B = \mathbf{p}_{blue}^* = (u_B, v_B), \quad C = \mathbf{p}_{green}^* = (u_C, v_C). \quad (9)$$

To improve robustness against noise and anti-aliasing artifacts, the LUT minimization can be refined by considering a local neighborhood around the detected point:

$$\tilde{\mathbf{p}}_c = \frac{1}{|\mathcal{N}(\mathbf{p}_c^*)|} \sum_{\mathbf{q} \in \mathcal{N}(\mathbf{p}_c^*)} \mathbf{q}, \quad (10)$$

where $\mathcal{N}(\mathbf{p}_c^*)$ denotes a small window centered at $\mathbf{p}_c^*$. This refinement reduces the effect of isolated noisy pixels and ensures sub-pixel accuracy.

3.3. Mask Validation and Correction

Points A and B are validated to lie within $\mathcal{M}_{gpt}$. If not, the closest mask pixel is found via expanding circle search:

$$A' = \arg \min_{\mathbf{p} \in \mathcal{M}_{gpt}} \|\mathbf{p} - A\|_2, \quad B' = \arg \min_{\mathbf{p} \in \mathcal{M}_{gpt}} \|\mathbf{p} - B\|_2 \quad (11)$$

This validation step ensures robustness against minor inaccuracies in GPT-generated landmark placement, preventing downstream errors in grasp computation.

The validation procedure is formalized as follows. For each landmark $\mathbf{p} \in \{A, B\}$, we check whether

$$\mathbf{p} \in \mathcal{M}_{gpt} \iff \mathcal{M}_{gpt}(\mathbf{p}) = 1. \quad (12)$$

If $\mathcal{M}_{gpt}(\mathbf{p}) = 0$, the landmark is corrected by finding the nearest pixel within the object mask:

$$\mathbf{p}' = \arg \min_{\mathbf{q} \in \mathcal{M}_{gpt}} \|\mathbf{q} - \mathbf{p}\|_2. \quad (13)$$

In practice, the nearest pixel is found efficiently via an expanding circular search. Let $\mathcal{C}(\mathbf{p}, r)$ denote the circle of radius r centered at $\mathbf{p}$:

$$\mathcal{C}(\mathbf{p}, r) = \{\mathbf{q} \in \mathbb{Z}^2 : \|\mathbf{q} - \mathbf{p}\|_2 \leq r\}. \quad (14)$$

The expanding search iteratively increases r until a pixel within $\mathcal{M}_{gpt}$ is found:

$$r^* = \min \{r \in \mathbb{R}^+ : \mathcal{C}(\mathbf{p}, r) \cap \mathcal{M}_{gpt} \neq \emptyset\}. \quad (15)$$

The corrected landmark is then

$$\mathbf{p}' = \arg \min_{\mathbf{q} \in \mathcal{C}(\mathbf{p}, r^*) \cap \mathcal{M}_{gpt}} \|\mathbf{q} - \mathbf{p}\|_2. \quad (16)$$

This correction is applied independently to A and B , yielding A' and B' . The wrist point C is not constrained to lie within the object mask, as the wrist is naturally outside the object region.

3.4. 2D Geometric Construction

Line ℓ_C is constructed through C parallel to AB , and perpendicular bisector ℓ_D of AB is constructed. Their intersection E is computed. Walking along $\overline{ED}$ from E to D , the first pixel inside $\mathcal{M}_{gpt}$ is identified as the grasp point T .

Let $\mathbf{A} = (u_A, v_A)$, $\mathbf{B} = (u_B, v_B)$, and $\mathbf{C} = (u_C, v_C)$ denote the 2D coordinates. The line ℓ_C through $\mathbf{C}$ parallel to $\mathbf{AB}$ is defined as

$$\ell_C : \mathbf{p}(t) = \mathbf{C} + t \cdot (\mathbf{B} - \mathbf{A}), \quad t \in \mathbb{R}. \quad (17)$$

The perpendicular bisector ℓ_D of $\mathbf{AB}$ is

$$\ell_D : \{\mathbf{p} \in \mathbb{R}^2 : \|\mathbf{p} - \mathbf{A}\|_2 = \|\mathbf{p} - \mathbf{B}\|_2\}. \quad (18)$$

Equivalently,

$$\ell_D : \mathbf{p}(s) = \mathbf{M}_{AB} + s \cdot \mathbf{n}_{AB}, \quad s \in \mathbb{R}, \quad (19)$$

where $\mathbf{M}_{AB} = \frac{\mathbf{A} + \mathbf{B}}{2}$ and $\mathbf{n}_{AB} = (v_B - v_A, -(u_B - u_A))$. The intersection point $\mathbf{E}$ of ℓ_C and ℓ_D is found by solving

$$\mathbf{C} + t \cdot (\mathbf{B} - \mathbf{A}) = \mathbf{M}_{AB} + s \cdot \mathbf{n}_{AB}. \quad (20)$$

This yields a 2×2 linear system:

$$\begin{bmatrix} u_B - u_A & -(v_B - v_A) \\ v_B - v_A & u_B - u_A \end{bmatrix} \begin{bmatrix} t \\ s \end{bmatrix} = \begin{bmatrix} u_{M_{AB}} - u_C \\ v_{M_{AB}} - v_C \end{bmatrix}. \quad (21)$$

The grasp point $\mathbf{T}$ is then found by walking along $\overline{ED}$ with step size δ :

$$\mathbf{T} = \mathbf{E} + k^* \cdot \delta \cdot \mathbf{d}_{ED}, \quad (22)$$

where

$$k^* = \min \{k \in \mathbb{N} : \mathcal{M}_{gpt}(\mathbf{E} + k \cdot \delta \cdot \mathbf{d}_{ED}) = 1\}. \quad (23)$$

3.5. Bounding-Box Transfer

For each mask $\mathcal{M}$, the bounding box center $\mathbf{V}$ and diagonal length L are computed. The scale ratio $r = L(\mathcal{M}_{gpt})/L(\mathcal{M}_{real})$ captures the relative scaling. For each point $\mathbf{P}$ in $\{\mathbf{A}, \mathbf{B}, \mathbf{T}\}$:

$$\mathbf{P}'' = \text{Project}_{\mathcal{M}_{real}} \left(\mathbf{V}_{real} + r \cdot \|\mathbf{P} - \mathbf{V}_{gpt}\|_2 \cdot \frac{\mathbf{P} - \mathbf{V}_{gpt}}{\|\mathbf{P} - \mathbf{V}_{gpt}\|_2} \right) \quad (24)$$

For a mask $\mathcal{M}$:

$$u_{\min} = \min_{(u,v) \in \mathcal{M}} u, \quad u_{\max} = \max_{(u,v) \in \mathcal{M}} u, \quad (25)$$

$$v_{\min} = \min_{(u,v) \in \mathcal{M}} v, \quad v_{\max} = \max_{(u,v) \in \mathcal{M}} v. \quad (26)$$

The bounding box center $\mathbf{V}$ and diagonal length L are

$$\mathbf{V} = \left(\frac{u_{\min} + u_{\max}}{2}, \frac{v_{\min} + v_{\max}}{2} \right), \quad (27)$$

$$L(\mathcal{M}) = \sqrt{(u_{\max} - u_{\min})^2 + (v_{\max} - v_{\min})^2}. \quad (28)$$

The scale ratio is

$$r = \frac{L(\mathcal{M}_{gpt})}{L(\mathcal{M}_{real})}. \quad (29)$$

The transferred point in the real image is

$$\mathbf{P}'' = \mathbf{V}_{real} + r \cdot \rho \cdot \mathbf{u}(\theta), \quad (30)$$

where $\rho = \|\mathbf{P} - \mathbf{V}_{gpt}\|_2$ and $\mathbf{u}(\theta) = \frac{\mathbf{P} - \mathbf{V}_{gpt}}{\|\mathbf{P} - \mathbf{V}_{gpt}\|_2}$. The projection operator is

$$\text{Project}_{\mathcal{M}_{real}}(\mathbf{P}) = \arg \min_{\mathbf{q} \in \mathcal{M}_{real}} \|\mathbf{q} - \mathbf{P}\|_2. \quad (31)$$

3.6. 3D Reconstruction and Grasp Frame Computation

Using the RGB-D camera intrinsic matrix $\mathbf{K}$:

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (32)$$

The back-projection to camera coordinates is:

$$\mathbf{P}_{camera} = \begin{bmatrix} (u - c_x) \cdot d / f_x \\ (v - c_y) \cdot d / f_y \\ d \end{bmatrix} \quad (33)$$

With camera extrinsics:

$$\mathbf{P}_{base} = \mathbf{R}_{camera}^{base} \mathbf{P}_{camera} + \mathbf{t}_{camera}^{base} \quad (34)$$

The 3D points $\mathbf{A}_{3D}$, $\mathbf{B}_{3D}$, and $\mathbf{T}_{3D}$ are obtained in the robot base frame. $\mathbf{K}$ is the midpoint of $\mathbf{A}$ and $\mathbf{B}$:

$$\mathbf{K} = \frac{\mathbf{A}_{3D} + \mathbf{B}_{3D}}{2}. \quad (35)$$

The vector from $\mathbf{T}_{3D}$ to $\mathbf{K}$ is

$$\mathbf{v}_{TK} = \mathbf{K} - \mathbf{T}_{3D}. \quad (36)$$

Let $\mathbf{u}_{AB} = \mathbf{A}_{3D} - \mathbf{B}_{3D}$. The component perpendicular to $\mathbf{v}_{TK}$:

$$\mathbf{u}_{AB}^\perp = \mathbf{u}_{AB} - \frac{\mathbf{u}_{AB} \cdot \mathbf{v}_{TK}}{\|\mathbf{v}_{TK}\|_2^2} \mathbf{v}_{TK}. \quad (37)$$

Then

$$\mathbf{L} = \mathbf{K} + \frac{1}{2} \mathbf{u}_{AB}^\perp, \quad \mathbf{M} = \mathbf{K} - \frac{1}{2} \mathbf{u}_{AB}^\perp. \quad (38)$$

The normal to the plane containing $\mathbf{L}$, $\mathbf{M}$, and $\mathbf{T}_{3D}$:

$$\mathbf{n} = (\mathbf{M} - \mathbf{L}) \times (\mathbf{T}_{3D} - \mathbf{L}). \quad (39)$$

The point $\mathbf{X}$ is

$$\mathbf{X} = \mathbf{K} + \hat{\mathbf{n}} \cdot \delta, \quad (40)$$

where $\hat{\mathbf{n}} = \mathbf{n} / \|\mathbf{n}\|_2$ and δ is a small offset.

3.7. Rule-Based Grasp Alignment

Let $\mathbf{Y}$ denote the 3D projection of the midpoint of $\mathbf{A}$ and $\mathbf{B}$:

$$\mathbf{Y} = \Pi_{3D} \left(\frac{\mathbf{A}'' + \mathbf{B}''}{2} \right). \quad (41)$$

The condition for Case 1 is

$$Y_z - K_z \geq -0.03, \quad (42)$$

and for Case 2 is

$$Y_z - K_z < -0.03. \quad (43)$$

For Case 1, the Z-axis is aligned parallel to $\mathbf{KX}$:

$$\mathbf{z}_{grasp} = \frac{\mathbf{X} - \mathbf{K}}{\|\mathbf{X} - \mathbf{K}\|_2}. \quad (44)$$

For Subcase 1a ($\|\mathbf{LM}\|_2 \geq 0.07$):

$$\mathbf{y}_{grasp} = \frac{\mathbf{K} - \mathbf{T}_{3D}}{\|\mathbf{K} - \mathbf{T}_{3D}\|_2}. \quad (45)$$

For Subcase 1b ($\|\mathbf{LM}\|_2 < 0.07$):

$$\mathbf{y}_{grasp} = \frac{\mathbf{M} - \mathbf{L}}{\|\mathbf{M} - \mathbf{L}\|_2}. \quad (46)$$

For Case 2:

$$\mathbf{z}_{grasp} = \frac{\mathbf{X} - \mathbf{K}}{\|\mathbf{X} - \mathbf{K}\|_2}, \quad \mathbf{y}_{grasp} = \frac{\mathbf{K} - \mathbf{T}_{3D}}{\|\mathbf{K} - \mathbf{T}_{3D}\|_2}. \quad (47)$$

In all cases:

$$\mathbf{x}_{grasp} = \mathbf{y}_{grasp} \times \mathbf{z}_{grasp}, \quad (48)$$

$$\mathbf{R}_{grasp} = [\mathbf{x}_{grasp} \quad \mathbf{y}_{grasp} \quad \mathbf{z}_{grasp}] \in SO(3). \quad (49)$$

The final homogeneous transformation matrix is, for Case 1:

$$\mathbf{T}_{grasp} = \begin{bmatrix} \mathbf{R}_{grasp} & \mathbf{K} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (50)$$

and for Case 2:

$$\mathbf{T}_{grasp} = \begin{bmatrix} \mathbf{R}_{grasp} & \mathbf{J} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (51)$$

where $\mathbf{J} = (\mathbf{T}_{3D} + \mathbf{K})/2$. The generalized threshold rule is

$$\text{Case} = \begin{cases} \text{Case 1a,} & Y_z - K_z \geq -\tau_z, \|\mathbf{LM}\|_2 \geq \tau_{LM}, \\ \text{Case 1b,} & Y_z - K_z \geq -\tau_z, \|\mathbf{LM}\|_2 < \tau_{LM}, \\ \text{Case 2,} & Y_z - K_z < -\tau_z, \end{cases} \quad (52)$$

with $\tau_z = 0.03$ m and $\tau_{LM} = 0.07$ m.

4. Experiments and Results

4.1. Experimental Setup

Franka Panda robot and Orbbec Gemini 2 RGB-D camera were used in Isaac Sim along with ROS2 integration, in a cluttered environment with groups of 6-9 objects (Fig. 2). The proposed method was validated on 130 diverse household objects including storage objects, decoration objects, toys, etc.

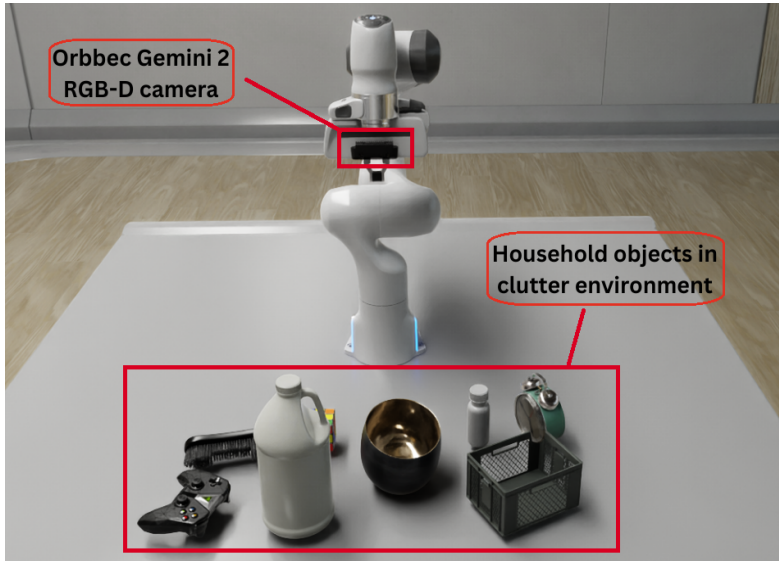


Figure 2: Isaac Sim and ROS2 setup for experimental verification

4.2. Results and Analysis

The complete experimental results across all 130 objects show that IMIGRASP achieves a success rate of 86.9% (113 successful grasps out of 130 attempts) with only a single attempt per object. Cylindrical and cuboid objects achieved approximately 93% success rate, the big reason was that GPT was itself very clear that from where to grasp this object. A complete per-object demonstration of the pipeline is shown in Fig. 3, covering PS Controller, Shoe Brush, Brass Bowl, Utility Jug, Plastic Container, Packer Bottle, and Alarm Clock.



Figure 3: IMIGRASP: Zero-shot 6-DOF grasp generation using generative AI for imagination guided pose estimation. Each row shows results for one of 7 objects i.e., PS Controller, Shoe Brush, Brass Bowl, Utility Jug, Plastic Container, Packer Bottle, and Alarm Clock. Columns show object mask in actual image, hand grasping image by GPT-5.6 Luna, object and hand masks in genAI image, genAI image with geometric construction lines, object in actual image with final mapped points A, B, T and successful grasp execution attempt by Franka Panda robot respectively.

Table 1: Success rate comparison with state-of-the-art methods.

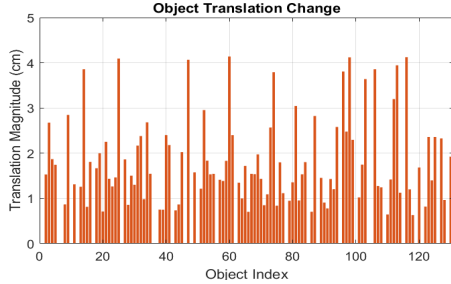
Method	Attempts	Success Rate (%)
Contact-GraspNet	Single	84.3
Cross-View Fusion	Multiple	96.0
IMIGRASP	Single	86.9

Table 1 presents the success rate comparison with state-of-the-art methods. In single attempt, it can be clearly seen that IMIGRASP has higher success rate. Fig. 3 shows complete

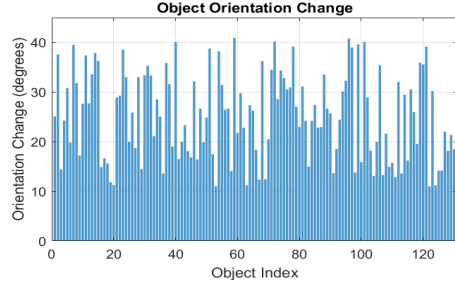
pipeline demonstration for 7 objects. Fig. 4 presents the displacement analysis during grasping approach. Maximum translation observed was 4.2 cm, while maximum orientation change was 41 degrees. Table 2 summarizes success and failure categories and their frequencies. Two representative failure cases are illustrated in Fig. 5 (Rubik’s Cube corner grasp that slipped) and Fig. 6 (Bowl bounding-box overflow and Oil Tin misidentified by GPT-5.6 Luna).

Table 2: Grasp performance and failure analysis across 130 objects.

Outcome	Occurrences	Percentage
Contact with other objects	81	62.3%
Minor movement of others	59	45.4%
Major movement of others	22	16.9%
Other objects fell	13	10.0%
GPT-5.6 Luna failure	10	7.7%
SAM3 wrong bbox	4	3.0%
SAM3 wrong segmentation	3	2.3%



(a)



(b)

Figure 4: Displacement and orientation error in all objects (a) Translation of objects from their original positions. (b) Orientation change of objects from their original orientations.



(a)



(b)

Figure 5: Rubik’s Cube failure case (a) GPT-5.6 Luna generated a grasp from the corner instead of the side. (b) The cube slipped under Franka Panda’s finger force, resulting in an unsuccessful grasp.

Of the 17 unsuccessful attempts, 10 were attributed to GPT-5.6 Luna failure, where the generated hand configuration was inappropriate for the object geometry or it grasped wrong object (Fig. 6b). The remaining 7 failures were attributed to SAM3, 4 due to SAM3 wrong bounding box (Fig. 6a), where the bounding box exceeded the object limits causing misplacement of mapped points and 3 failures due to SAM3 segmenting wrong object. In cluttered environments with groups of 6-9 objects, 81 successful grasps occurred with the object touching

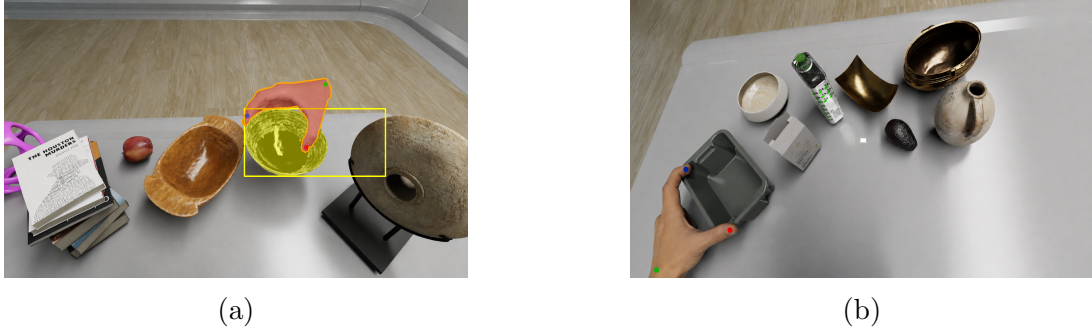


Figure 6: Failure case examples. (a) Bowl: SAM3 bounding box exceeds object limits causing misplacement of mapped points. (b) Oil Tin: GPT-5.6 Luna incorrectly generated a grasping configuration for a plastic container instead of the oil tin.

other objects. Among these, 59 resulted in minor movement of surrounding objects, 22 caused major movement, and 13 objects fell during the process (Table 2).

4.3. Comparative Analysis

A key advantage of the proposed zero-shot approach is its favorable success rate, space complexity, or zero prior training requirement, depending on the method compared, relative to training-based and retrieval-based approaches. Training-based methods like Contact-GraspNet [10] require 40 hours of GPU training on 17 million simulated grasps, while Cross-View Fusion [16] requires training on the large-scale GraspNet-1B dataset. Retrieval-based methods like GRIM [1] require memory construction (7 minutes per item) and online semantic alignment (10 seconds per grasp) during inference. In contrast, IMIGRASP operates with no training phase. IMIGRASP achieves near constant time complexity $\mathcal{O}(1)$ with respect to the number of objects or training examples. The space complexity is equally minimal $\mathcal{O}(1)$. In terms of success rate, IMIGRASP (86.9%) outperforms Contact-GraspNet’s single-attempt rate (84.3%). Conversely, Cross-View Fusion achieves a higher success rate (96.0%), but this comes at the cost of allowing multiple attempts and requiring more VRAM and training data.

5. Conclusion

IMIGRASP eliminates the need for training data, hand pose estimation networks, or object-specific tuning, making it suitable for applications where data collection is impractical. Its lower computational footprint and competitive success rate enable real-time deployment on robotic systems in warehouses and household manipulation tasks. The results validate that generative AI models, trained on internet-scale visual data, can serve as imagination priors for robotic grasping, reducing reliance on expensive training pipelines.

Acknowledgements

The author gratefully acknowledges the Department of Electrical Engineering, Pakistan Institute of Engineering and Applied Sciences (PIEAS), for providing research guidance and academic support throughout this work. No specific funding was received from the Department of Electrical Engineering or any other agency for this research.

The author also acknowledges the use of generative AI tools for assistance with LaTeX code generation and with the development of execution codes used in the experiments.

References

- [1] Shailesh, A. Raj, N. Kumar, P. Shukla, A. Melnik, M. Beetz, and G. C. Nandi, “GRIM: Task-oriented grasping with conditioning on generative examples,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2024.
- [2] H. Li, W. Mao, W. Deng, C. Meng, R. Zhang, F. Jia, T. Wang, H. Fan, H. Wang, and X. Deng, “SegGrasp: Zero-shot task-oriented grasping via semantic and geometric guided segmentation,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2025.
- [3] L. Heng, J. Xu, Y. Wang, X. Li, M. Cai, Y. Shen, J. Zhu, G. Ren, and H. Dong, “Imagine2Act: Leveraging object-action motion consistency from imagined goals for robotic manipulation,” *arXiv preprint*, 2025.
- [4] Y. Ye, X. Li, A. Gupta, S. De Mello, S. Birchfield, J. Song, S. Tulsiani, and S. Liu, “Affordance diffusion: Synthesizing hand-object interactions,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023.
- [5] J. Achiam et al., “GPT-4 technical report,” *arXiv preprint arXiv:2303.08774*, 2023.
- [6] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, “Hierarchical text-conditional image generation with CLIP latents,” *arXiv preprint arXiv:2204.06125*, 2022.
- [7] OpenAI, “GPT-4: Generative Pre-trained Transformer 4,” 2023.
- [8] A. Kirillov et al., “Segment anything,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023.
- [9] N. Ravi et al., “SAM 2: Segment anything in images and videos,” *arXiv preprint arXiv:2408.00714*, 2024.
- [10] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, “Contact-graspnet: Efficient 6-DoF grasp generation in cluttered scenes,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pp. 13438-13444, 2021.
- [11] B. Han, Y.-W. Chao, E. Coumans, C. Eppner, B. Sundaralingam, J. Deng, S. Birchfield, and A. Murali, “GraspGen-X: Cross-embodiment 6-DOF diffusion-based grasping,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2026.
- [12] E. Xiang, H. Ma, X. Ma, Z. Liu, and D. Huang, “GraspLDP: Towards generalizable grasping policy via latent diffusion,” *arXiv preprint*, 2025.
- [13] D. Wu, X. Wei, H. Wang, Q. Dong, Y. Li, Q. Wuwu, M. Lu, and W. Zhao, “GraspFoM: Towards reconstruction-driven robotic grasping with 3D foundation priors,” *arXiv preprint*, 2026.
- [14] H. Akada, J. Pérez, E. Aksan, V. Choutas, C. Romero, A. Garcia-Garcia, V. Golyanik, C. Theobalt, and T. Beeler, “DCGrasp: Distance-aware controllable grasp generation,” *arXiv preprint*, 2025.
- [15] J. Lee, E. Park, and M. Cho, “DextER: Language-driven dexterous grasp generation with embodied reasoning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2025.
- [16] K. Zhu, H. Jiang, J. Qian, and J. Xie, “A cross-view fusion framework for robust 6-DoF grasp pose estimation,” *arXiv preprint*, 2025.

- [17] H.-S. Fang, C. Wang, H. Fang, M. Gou, J. Liu, H. Yan, W. Liu, Y. Xie, and C. Lu, “AnyGrasp: Robust and efficient grasp perception in spatial and temporal domains,” *IEEE Trans. Robot.*, vol. 39, no. 5, pp. 3929-3945, 2023.
- [18] G. Li, D. Sun, L. Sevilla-Lara, and V. Jampani, “One-shot open affordance learning with foundation models,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 3086-3096, 2024.
- [19] S. Li, S. Bhagat, J. Campbell, Y. Xie, W. Kim, K. Sycara, and S. Stepputtis, “Shapegrasp: Zero-shot task-oriented grasping with large language models through geometric decomposition,” *arXiv preprint arXiv:2403.18062*, 2024.
- [20] Y.-L. Wei, J.-J. Jiang, C. Xing, X.-M. Wu, H. Li, M. Cutkosky, and W.-S. Zheng, “Grasp as you say: Language-guided dexterous grasp generation,” *arXiv preprint arXiv:2405.19291*, 2024.
- [21] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, “Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection,” *Int. J. Robot. Res.*, vol. 37, no. 4-5, pp. 421-436, 2018.
- [22] J. Mahler et al., “Dex-Net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics,” in *Proc. Robot.: Sci. Syst.*, 2017.
- [23] S. Caldera, A. Rassau, and D. Chai, “Review of deep learning methods in robotic grasp detection,” *Multimodal Technol. Interact.*, vol. 2, no. 3, p. 57, 2018.
- [24] R. Wang, J. Zhang, J. Chen, Y. Xu, P. Li, T. Liu, and H. Wang, “Dexgraspnet: A large-scale robotic dexterous grasp dataset for general objects based on simulation,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pp. 11359-11366, 2023.
- [25] T. Liu, Z. Liu, Z. Jiao, Y. Zhu, and S.-C. Zhu, “Synthesizing diverse and physically stable grasps with arbitrary hand structures using differentiable force closure estimator,” *IEEE Robot. Autom. Lett.*, vol. 7, no. 1, pp. 470-477, 2021.
- [26] H. Luo, W. Zhai, J. Zhang, Y. Cao, and D. Tao, “Learning affordance grounding from exocentric images,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 2252-2261, 2022.
- [27] M. Liu, Y. Zhu, H. Cai, S. Han, Z. Ling, F. Porikli, and H. Su, “Part-slip: Low-shot part segmentation for 3d point clouds via pretrained image-language models,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 21736-21746, 2023.
- [28] A. Abdelreheem, I. Skorokhodov, M. Ovsjanikov, and P. Wonka, “SATR: Zero-shot semantic segmentation of 3d shapes,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pp. 15166-15179, 2023.
- [29] A. Radford et al., “Learning transferable visual models from natural language supervision,” in *Proc. Int. Conf. Mach. Learn. (ICML)*, pp. 8748-8763, 2021.
- [30] C. Finn, I. Goodfellow, and S. Levine, “Unsupervised learning for physical interaction through video prediction,” in *Adv. Neural Inf. Process. Syst.*, 2016.